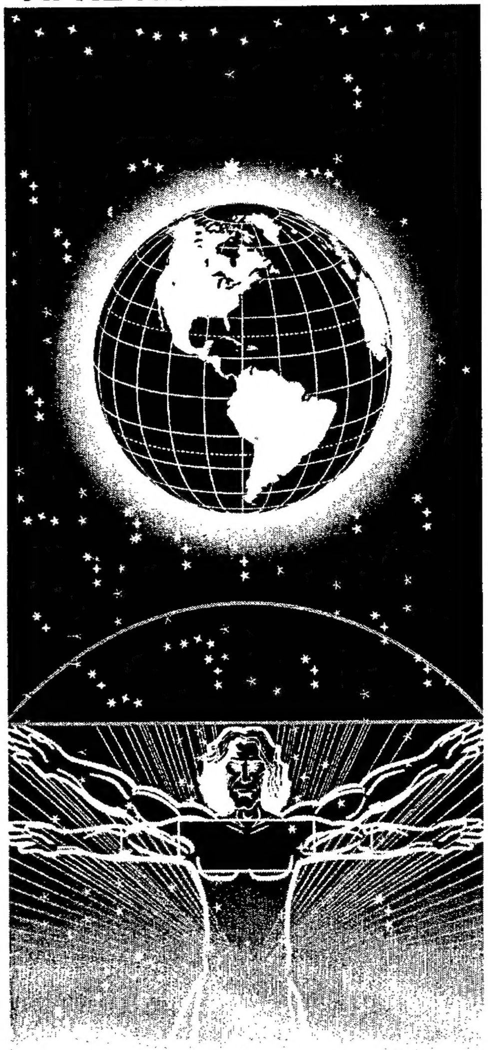
AFRL-HE-WP-SR-2000-0009



UNITED STATES AIR FORCE RESEARCH LABORATORY

A NEURAL NETWORK MODEL FOR HUMAN WORKLOAD SIMULATION IN COMPLEX HUMAN-MACHINE SYSTEM

Celestine A. Ntuen

DEPT. OF INDUSTRIAL & SYSTEM ENGINEERING
NORTH CAROLINA A&T STATE UNIVERSITY
1601 EAST MARKET STREET
GREENSBORO NC 27411

Robert Li

DEPT. OF ELECTRICAL ENGINEERING
NORTH CAROLINA A&T STATE UNIVERSITY
1601 EAST MARKET STREET
GREENSBORO NC 27411

DECEMBER 1999

FINAL REPORT FOR THE PERIOD MAY 1998 TO OCTOBER 1999

20010326 042

Approved for public release; distribution is unlimited.

Human Effectiveness Directorate
Crew System Interface Division
2255 H Street
Wright-Patterson AFB OH 45433-7022

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
Dec 1999

3. REPORT TYPE AND DATES COVERED
Final Report, May 1998 to October 1999

4. TITLE AND SUBTITLE
A Neural Network Model for Human Workload Simulation in a Complex Human-Machine System

5. FUNDING NUMBERS
C F41624-98-1-0002
PE 62202F
PR 7184
TA 14
WU AT

6. AUTHOR(S)

Celestine A. Ntuen*
Robert Li*

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

*North Carolina A&T State University
1601 East Market Street
Greensboro NC 27411

8. PERFORMING ORGANIZATION REPORT NUMBER
IHMS-99-003/USAF

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory
Human Effectiveness Directorate
Crew System Interface Division
Air Force Materiel Command
Wright-Patterson AFB, OH 45433-7022

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-HE-WP-SR-2000-0009

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The overall goal of this study is to develop neural network models for analysis of electroencephalogram (EEG) data and use the results obtained to classify the level of mental workload experienced by humans during task processing. The study uses EEG data on piloting tasks from the STORM (Simulator for Tactical Operations Research and Measurement) experiments performed at the Cognitive Assessment Laboratory of the Human Effectiveness Directorate at Wright-Patterson AFB. Comparisons of classical backpropagation neural networks (CBNN) and resilient backpropagation neural networks (RBNN) were conducted. The RBNN performed 50% faster in deriving a model for cognitive load with a marginal decrease in classification accuracy over the CBNN. The results indicate that the neural network model can successfully classify mental workload states at an average rate of 83%. The results obtained indicate that neural network models can be used to automate the classification of human mental workload based on EEG signal data.

14. SUBJECT TERMS

Neural networks, Workload, Man-machine systems

15. NUMBER OF PAGES
55

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UNLIMITED

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

ASQ01-03-0307

FINAL TECHICAL REPORT #: IHMS-99-003/USAF

A NEURAL NETWORK MODEL FOR HUMAN WORKLOAD SIMULATION IN A COMPLEX HUMAN-MACHINE SYSTEM

Performance Period: May 1998 - October 1999

Prepared for

U.S. Air Force
Crew System Laboratory
AL/DFPH, 2255 Hst
Wright Patterson Air Force Base, OH 04533-7022
Technical Monitor: Dr. Gary B. Reid
Grant #: USAF 41624-98-1-0002

Submitted by

Dr. Celestine A. Ntuen, PI
Department of Industrial & Systems Engineering

Dr. Robert Li
Department of Electrical Engineering

North Carolina A&T State University
1601 East Market Street
Greensboro, NC 27411

December 1999

TABLE OF CONTENTS

LIST OF FIGURES .... Page No.

LIST OF TABLES .... Page No.

ACKNOWLEDGEMENTS

We owe our profound gratitude to Mr. Chris A. Russell for his technical advisement during the early part of this project. He also made available the computer code for Ruck saliency that was used in significant feature extraction. We thank Dr. Gary B. Reid for his patience and for providing us with information on STORM and its experimental data.

We must acknowledge the significant contributions of our graduate students: Hilda Goins, a Ph.D Student in Electrical Engineering; and Chen Ling, a Master Student in Industrial and Systems Engineering. This project has provided the students the research background for their dissertation and thesis.

SUMMARY

As human control tasks in human-machine systems get automated, the human role in the system is gradually shifting to that of a supervisory control. In this new role, the human is expected to engage more on automation monitoring. Hence, mental workload is expected to be the dominant measure of human performance.

Mental workload is a function of information processing by the human brain. Studies in neurophysiology have indicated that EEG (Electroencephalogram) provides one way to measure mental workload.

EEG data consist of a transient of voltage oscillations in the brain that are recorded using the international standard of neurophysioloical and biomedical instrument conditions. While the use of EEG has an advantage of providing a non-invasive measure of mental workload, the quantitative methods for analyzing and interpreting EEG data remains an important field of study. An approach advocated and adopted recently to solving this problem is the use of neural network.

The overall goal of this investigation is to develop neural network models for analysis of EEG data and use the results obtained to classify the level of mental workload experienced by humans during task processing.

The study uses EEG data on piloting task from the STORM (Simulator for Tactical Operations Research and Measurement) experiments performed at the Crew System Integrated Laboratory of the Human Effectiveness Directorate at Wright Patterson Air Force Base.

The EEG data was grouped by using 80% overlap at 10-sec. interval. Fast Fourier Transform (FFT) filter was used to process the desired features needed for the neural network models. The feature data was analyzed with the classical backpropagation neural network (CBNN) and resilient backpropagation (RBNN) neural network. The pilot tasks were at three levels: forward, backward, and dual processing, respectively. Six levels of task complexity were investigated. Generically, a task complexity corresponds to an aircraft speed. The speeds analyzed range from comfortable workload (215 knots) to an obvious overload condition (600 knots). The six speed levels used are 215, 325, 380, 435, 490, and 600 knots, respectively. The speed levels are assumed to simulate the pilot workload in a progressive perception scale increasing from very low workload to overload.

By using individual subject data on workload levels of twelve pilots, the RBNN was considered 50% faster in processing the necessary neural network information. This was measured by the mean number of epochs. For forward piloting task, RBNN took 173 epochs versus 369 by CBNN; for reverse piloting task, RBNN took 233 epochs versus 404 by CNN; and for dual piloting task, RBNN took 177 epochs versus 749 by CBNN. The main pay off derived from this result is the reduction in computer processing time. When percentage of classification accuracy was used, the CBNN marginally performed

better than RBNN as shown in Table I below. However, there was no significant difference in performance classification when pilots perform dual processing tasks.

Table 1: Mean classification performance (as % accuracy)

Workload Classification	Piloting Task					
	Forward		Reverse		Dual	
	CBNN	RBNN	CBNN	RBNN	CBNN	RBNN
Aircraft velocity	84.01	82.34	83.19	81.56	79.23	78.33
Pilots	84.14	82.46	83.83	82.2	79.29	78.38

The results obtained indicate that neural network models can be used to automate the classification of human mental workload state based on EEG signal data. Our results further indicate that the neural network models used can successfully classify mental workload states at an average rate of 83%. We believe that the classification rate can be further improved by the use of fuzzy classification techniques. However, this was not pursued due to lack of subjective data within and across the workload categories. It certainly merits further investigation.

1. INTRODUCTION

Workload classification problem has been, and will continue to provide the theoretical basis for work design and assignment between human and machines (Eggemeir & Reid, 1986). Unfortunately, in practice, such standard classification metrics like Cooper-Harper (1969), SWAT and NASA-TLX (Bortolussi, Kantowitz, & Herst, 1986) are often used solely for determining the levels of the human performance. One important reason for not using subjective workload metric as a classifier model for human-machine task allocation is that, in general, subjective measures are rarely stable, insensitive to task changes, and too local for generalization to diverse contexts. This is a problem this project seeks to address.

In the past twenty years, due to the developments in biomedical instrumentation technologies, interest in using human brain state activities to model workload has increased (Biferno, 1985; Doyle, Ornstein & Galin, 1974; Etterna & Zielhuis, 1971; Moser & Annon, 1986). The main assumption is that if the brain-state classifier can be found, then it can be used to detect correlation between brain signals and the level of workload due to information processing requirement imposed on the operator (Bauer, Goldstein & Stern, 1987; Beatty, 1982; Horst, Munson & Ruchklin, 1984).

Although the use of EEG has provided a non-invasive and robust approach to mental workload modeling (Russell & Wilson, 1996), the methods for translating EEG data and matching them to the subjective rating of cognitive effort as perceived by the human operator during task processing still remains a difficult task. An approach advocated and adopted recently to solving this problem is the use of neural network (Gevins, 1988; Russell & Wilson, 1996; Fukuda, Tsuji, & Kaneko, 1996 Anderson, Derulapalli & Stotz, 1995; Kloppel, 1994; Greene et al; 1996; Fukuda, Tsuji, & Kaneko, 1996).

Before elaborating on our approach, a synopsis of approaches to EEG applications to mental workload is presented.

2. RELEVANCE OF PAST STUDIES

In the past twenty years, psychophysiologists and neural scientists have advocated the use of EEG signals as a more realistic approach to understanding human mental workload during task performance (see, e.g., Kramer, 1991; Wilson & Eggemeier, 1991; Klimesh, Schimke & Pfurtscheller, 1993).

There are two related topics that are important in these studies. These are

(a) Physiological data acquisition using EEG apparatus.

EEG data consist of a transient of voltage oscillations in the brain that are recorded on areas corresponding to the international standard of neurophysiological and biomedical instrumentation conditions (Gasper, 1958). Usually, EEG data consist of a continuous

stream of pulsating Event-related Brain Potentials (ERP). ERPs are viewed as a sequence of separate but sometimes temporally overlapping components which are influenced by some combinations of the physical parameters of the eliciting stimuli and psychological constructs such as varieties of attention, memory or response processes (Humphrey & Kramer, 1993; Fruhstorfer & Bergstorm, 1969).

EEG signals contain noise, mostly from the surface electrodes, electrical inference, and other muscular activities. If this noise can be removed, several mental states corresponding to human activities during task performance can be distinguished by recognizing variations and patterns in EEG data (Givens, et al, 1979; Hancock, Meshikati & Robertson, 1985; Kramer & Strayer, 1988; Jasen & Dawnat, 1989; Wu, Ifeachor, Allen, Wimalaranta & Hudson, 1997).

(b) Classification of EEG Data

The major concern in EEG studies is how to use the EEG data in real-time to measure momentary fluctuations in mental activities (Lowe, 1998). One approach of solving this problem is to divide the EEG data into time segments and model the mental activities as time series, autoregressive (AR) models (Shiao-Liu, Yi-Jean & Cheng-Yuan, 1993; Keirn & Aunon, 1990). In this case, each data segment denoted as $\hat{x}(t)$ at time t is a linear combination of the values at the g preceding instants. This is defined by equation (1) as:

$$\hat{x}(t) = \sum_{k=1}^{g} a_k^g \times (t - k\Delta t) \qquad (1)$$

Here, Δt is the sampling time and g is called the AR model's order. The AR coefficients, a_k^g (k=1, 2, ..., g) are usually obtained by minimizing the summed square error defined by

$$SQE = \sum_{t=0} [x(t) - \hat{x}(t)]^2 \qquad (2)$$

Another approach to EEG data classification is the use of linear discriminate analysis (Aasman, Mulder & Mulder, 1987; Horst & Donchin, 1980). The discriminate procedure computes linear and quadratic functions for classifying observations into two or more groups on the basis of one or more numeric variables. The classification criterion is based on either the individual within-group covariance matrices or the pooled covariance matrix; it also takes into account the prior probabilities of the groups. Each observation is placed in the class from which it has the smallest generalized square distance.

The third approach and most recently advocated is the use of neural network (Gevins, 1988; Wilson & O'Donnell, 1988; Russell & Wilson, 1996; Anderson, Devulapalli & Stolz, 1995; Greene et al, 1996; Tsoi, So, & Sergejew, 1994; Wang, He & Chen, 1999).

The theory of neural networks suggested by McCullouch and Pitts (1943) is that the behavior and performance of a biological organism can be modeled by using its sensory

receptors and data from responsive actions. This concept was equated to transducers that process electronic information mostly by converting one form of energy into another. This paradigm analogically resembles the energy exchange during task execution by the human operator. This principle developed by McCuloch and Pitts (1943) works as follows: if we start with some (n) receptor neurons $r_1, r_2, \ldots, r_n$ and some (m) effectors $e_1, e_2, \ldots, e_m$, conforming to any pre-assigned input-output prescription, we can assemble these three neurons in a manner that enables us to obtain a single value of the system behavior. Recent work by Cohen et al, (1990) confirms the robustness of using neural networks to study human performance such as attentional processes involving color perception, color naming, and work recognition. Grossberg (1982) in most of his works, notes that the implicit parallelism of internodal activities, the competitive nature of excitatory and inhibitory information flows, the chemical equation induced by stress, etc., all provide the neural substrates for a mass balance equation similar to a complex engineering system. Again, this assures us that human performance models can be formulated as time-dependent dynamic (differential) equations. Workload, a basic parameter of performance, characteristically fits into the Grossberg's neural-dynamic paradigms.

A typical neural network model consists of input nodes, hidden layer levels and nodes, and output nodes. Although knowledge representation is sequential across the levels, the model processing is parallel, similar to how humans process information. Classical workload metrics does not, in principle, differentiate these dimensions in human performance analysis and can not model this parallel-processing behavior.

3. THE CURRENT STUDY

3.1 Caveat

This study was undertaken to understand the relationship between the increase velocity of an aircraft and a pilot's ability to effectively perform given tasks. Levels of mental workload were defined according to the associated speed in the simulated cockpit. Simultaneous data from multiple electroencephalograph (EEG) channels was recorded and used as input features to a backpropagation feedfroward neural network for classifying the pilot's cognitive workload. A set of classification results is performed for 12 subjects.

Studies on EEG applications to workload modeling have been performed in the domain of piloting and flight tasks. For example, Kramer , Sirevaag, & Braun, 1987; Lindholm, Cheathan, Koriath, & Longridge, 1984; Skelly, Purvis & Wilson, 1987; Roscoe, 1975; Itoh, Hayashi, Tskui & Saito, 1987; Greene et al, 1996; Wilson, 1973; Sterman et al, 1992; Brookings, Wilson & Swain, 1996. These studies confirm the robustness of EEG data as a good estimator of the human mental state, and neural network as the best classifier of those states based on the desired workload classes.

3.2 Neural Network Models

In this study, an artificial neural network is used to classify the pilot's workload data into six classes. A neural network classifier works by synaptic modification algorithms that allow an arbitrarily connected network to develop an internal structure appropriate for a particular task, such as classifying. In solving classification problems of this nature, a multi-layer feedforward neural network trained by backpropagation algorithm is utilized. Upon learning, the network is expected to correctly classify similar data. The workload-sensitive features of the continuous EEG were identified and extracted for further study. Effective automated means for classifying the features according to task difficulty (indicated by trial speed) were developed and tested. The goal was to test the ability of a neural network model, namely backpropagation, by analyzing the accuracy of prediction of the workload (indicated by the trial speed).

3.2.1 Classical Backpropagation Neural Network (CBNN)

The neural network model used in this study is a multi-layer network (MLN) operating with a feedforward backpropagation algorithm (Greene, Bauer, Kabrisky, Rogers, Russell, & Wilsom, 1996). Figure 1 is used to depict a typical MLN architecture.

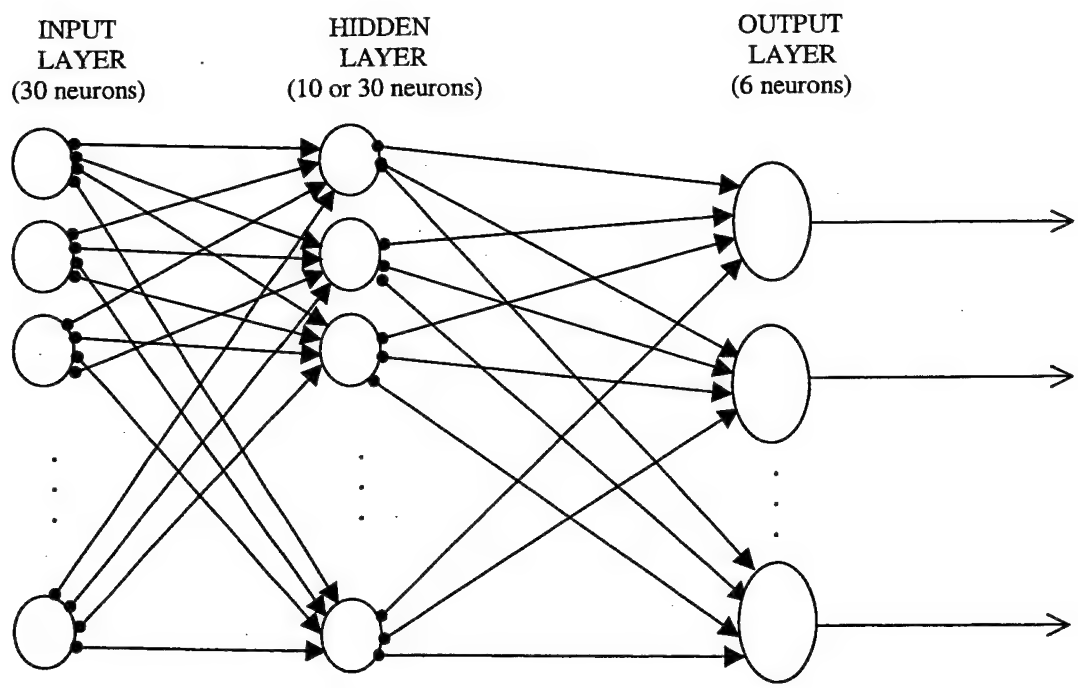


Figure 1. The architecture of the neural network

The neural network model is trained by the Generalized Delta Rule (GDR). The GDR can be treated as a gradient descent process that converges to the basin of least mean square (LMS) error between the actual and target outputs. If the outputs produced are identical to the target value, then no weight change takes place. Otherwise, the LMS

error between the actual and target value is propagated backward through the network to update the synaptic connection weight.

The normalized input is passed through a set of weights initialized to random values, and the inner product between the input and weights is calculated using dot product. That result is then passed through an activation function. The activation function used in this experiment for all layers was a logistic sigmoid. This function is

$$f(a)=\frac{1}{1+e^{-a}} \tag{3}$$

Equation (3) limits the output to a certain range of values. The output node is computed from the weighted sum of the inputs to the node from the previous layer. The weight sum of inputs is termed the activation and is denoted by a. Each node has an associated activation function, and associated activation function f(a). The output of the node is the result of applying this activation function. Hence the output in the l th layer, $y_i^l = f\left(a_i^l\right)$.

Two other activation function generally chosen are:

(i) Linear activation

$$f(a)=a \tag{4}$$

(ii) Hyperbolic tangent or symmetric sigmoid activation:

$$f(a)=\tanh(a) \quad = \quad \frac{e^a - e^{-u}}{e^a + e^{-a}} \tag{5}$$

For the model in Figure 1

$$y_i^l = f\left(\sum_{j=0}^{n(l-1)} a_{ij}^l\right) \tag{6}$$

By using equation (3)

$$y_i^l = \frac{1}{1+e^{\left(-\sum_{j=0}^{n(l-1)} a_{ij}^l\right)}} \tag{7}$$

$$a_{ij}^l = w_{ij}^l y_j^{l-1} + b_j^l \tag{8}$$

where

n_l :the number of nodes in layer l

a_{ij}^l :the linear connection function between layer 1 and l-1

y_j^l :the output of jth node in layer l

b_j :the bias input to jth node of layer l

The network is trained on a collection of input-output patterns, with an error backprogapation method. A pattern is presented at the bottom layer of the network, after which the network produces an output pattern. This phase is known as forward pass. The actual patterns of the network in the final layer are compared with their layer values for the given pattern. The differences between the network output and actual (desired) output defines the error function similar to ε_p defined in equation (9). The gradient search method error is often used to minimize such error function (Sun, Ryan, Dahl, Iyengar and Sclabass; 1993). A typical error function is

$$\varepsilon_p = 0.5 \sum_{p=1}^{P} \sum_{j}^{J} \left[t_{pj} - Y_{pj} \right]^2 \tag{9}$$

Where t_{pj} and Y_{pj} are the target and the current output values for pattern p, respectively, p is the number of patterns, and j is the number of output nodes in the output layer.

3.2.2 Learning in Backpropagation Neural Network

The learning process in backpropagation algorithm depends in part on the changing error gradients coming from different weights and the methods used in update the weights. The learning is based on the delta rule of supervised learning (Peters, Pfurstscheller & Flyvbjerg, 1988; McClelland & Rumelhart, 1986). The batch propagation weight update is a form of gradient descent defined by

$$W_{ij}^{l} = W_{ij}^{l-1} + \Delta W_{ij} \tag{10}$$

ΔW_{ij} can take various forms (Xiao, Yang & Zhou, 1977)

$$\Delta W_{ij}^{n} = -\eta \sum_{p=1}^{P} \frac{\partial \varepsilon_p}{\partial W_{ij}} \tag{11}$$

Where η is the step size or learning rate and n is the number of iterations.

$$\Delta W_{ij}^{n} = -\eta \sum_{p=1}^{P} \frac{\partial \varepsilon_p}{\partial W_{ij}} + \alpha \Delta W_{ij}^{n-1} \tag{12}$$

$$\Delta W_{ij}^{n} = -(1-\alpha)\eta \sum_{p=1}^{P} \frac{\partial \varepsilon_p}{\partial W_{ij}} + \alpha \Delta W_{ij}^{n-1} \tag{13}$$

In equation (13)

$$0 \le \alpha \le 0.5, \quad \eta = \frac{\alpha}{1-\alpha}$$

Backpropagation neural networks support nonlinear input-output relations, and they are considered useful in EEG pattern classification (Jung & Makeig, 1994; Kloppel, 1994; Morton, Turney, et al; 1991; Bird, Newton, et al; 1978; Wu, Ifeachor, Allen & Hudson, 1994; Pardey, Roberts & Tarassenko, 19XX; Fukuda, Tsuji & Kaneko, 1995; Hiraiwa, Simohara & Tokunaga, 1989; Peltoranta & Pfurtscheller, 1994; Reddy6 & Rao, 19XX).

With a backpropagation network, a segment of interaction traces can be presented to the input layer of a back-propagation network, and the identification or the classification of the segment can be produced at the output layer of the back-propagation network. The supervised learning of the back-propagation network is based on examples of known user patterns.

Because backpropagation networks often out perform conventional linear and polynomial predictive and statistical techniques in representing nonlinear input-output relations (Ye, 1997), these networks are also valuable in static user modeling. The robustness to noise and, the parallel processing are some additional merits of backpropagation networks compared to traditional predictive and statistical techniques.

Moreover, if a backpropagation network is trained in an autoassociative fashion, it stores user patterns in it structure. When some partial or noisy cue of a stored user pattern is presented to the input layer of the network, the output layer of the network recalls the complete user pattern. Hence, back-propagation networks can be utilized for user pattern storage and retrieval to complete partially recognized user patterns. Retrieved user patterns can then be used to predict user sequences of actions for automatic execution.

3.2.3 Resilient Backpropagation Neural Network (RBNN)

Riedmiller and Braum (1993) proposed the resilient back-propagation neural network (RBNN) as an improvement to the CBNN, which has the tendency of getting "trapped" into local minima during gradient search. Riedmiller and Braum refer to their model as RPROP, which means for Resilient PROPagation.

The RPROP (or RBNN as used here) is an efficient learning adaptation of the weight step based on local gradient information. The following procedure summarizes the RBNN working algorithm (Riedmiller & Braum, 1993):

For all weights and biases{

$$\text{if}\left(\frac{\partial\varepsilon}{\partial W_{ij}}(t-1)*\frac{\partial\varepsilon}{\partial W_{ij}}(t)>0\right) then \,\{$$

$$\Delta_{ij}(t)=\text{minimum}\left(\Delta_{ij}(t-1)*\eta^{+},\Delta_{\max}\right)$$

$$\Delta W_{ij}(t)=-sign\left(\frac{\partial\varepsilon}{\partial W_{ij}}(t)\right)*\Delta_{ij}(t)$$

$$W_{ij}(t+1) = W_{ij}(t) + \Delta W_{ij}(t)$$

}

else if $\left(\frac{\partial \varepsilon}{\partial W_{ij}}(t-1) * \frac{\partial \varepsilon}{\partial W_{ij}} \right)(t) < 0$) *then* {

$$\Delta_{ij}(t) = \text{maximum}\left(\Delta_{ij}(t-1) * \eta^{-}, \Delta_{\min}\right)$$

$$W_{ij}(t+1) = W_{ij}(t-1) - \Delta W_{ij}(t-1)$$

$$\frac{\partial \varepsilon}{\partial W_{ij}}(t) = 0$$

}

else if $\left(\frac{\partial \varepsilon}{\partial W_{ij}}(t-1) * \frac{\partial \varepsilon}{\partial W_{ij}}(t) = 0 \right)$ *then* {

$$\Delta W_{ij(t)} = -\, sign\left(\frac{\partial \varepsilon}{\partial W_{ij}}(t) \right) * \Delta_{ij}(t)$$

$$W_{ij}(t+1) = W_{ij}(t) + \Delta W_{ij}(t)$$

}

}

In the above procedure, the learning rule, indicated by its individual update-value Δ_{ij} is:

$$\Delta_{ij}^{(t)} = \begin{cases} \eta^{+} * \Delta_{ij}^{(t-1)} &, \ if \ \frac{\partial E^{(t-1)}}{\partial W_{ij}} * \frac{\partial E^{(t)}}{\partial W_{ij}} > 0 \\ \eta^{-} * \Delta_{ij}^{(t-1)} &, \ if \ \frac{\partial E^{(t-1)}}{\partial W_{ij}} * \frac{\partial E^{(t)}}{\partial W_{ij}} > 0 \\ \Delta_{ij}^{(t-1)} &, \ else \\ where \ 0 < \eta^{-} < 1 < \eta^{+} & \end{cases} \tag{14}$$

Every time the partial derivative of the corresponding weight W_{ij} changes it sign, which indicates that the last update was too big and the algorithm has jumped over a local minimum, the update-value Δ_{ij} is decreased by the factor η^{-}. If the derivative retains its sign, the update-value is slightly increased in order to accelerate convergence in shallow regions.

Once the update-value for each weight is adapted, the weight-update itself follows a very simple rule: if the derivative is positive (increasing error), the weight is decreased by its update-value, if the derivative is negative, the update-value is added:

$$\Delta W_{ij}^{(t)} = \begin{cases} -\Delta_{ij}^{(t)} & , \ if \ \dfrac{\partial E^{(t)}}{\partial W_{ij}} > 0 \\ +\Delta_{ij}^{(t)} & , \ if \ \dfrac{\partial E^{(t)}}{\partial W_{ij}} > 0 \\ 0 & , \ else \end{cases} \tag{15}$$

$$W_{ij}^{(t+1)} = W_{ij}^{(t)} + \Delta W_{ij}^{(t)} \tag{16}$$

However, there is one exception: if the partial derivative changes sign, i.e. the previous step was too large and the minimum was missed, the previous weigh-update is reverted.

$$\Delta W_{ij}^{(t)} = -\Delta W_{ij}^{(t-1)} \quad , if \ \frac{\partial W^{(t-1)}}{\partial W_{ij}} * \frac{\partial W^{(t)}}{\partial W_{ij}} < 0 \tag{17}$$

Due to that 'backtracking' weight-step, the derivative is supposed to change its sign once again in the following step. In order to avoid a double punishment of the update-value, there should be no adaptation of the update-value in the succeeding step. In practice this can be done by setting $\dfrac{\partial \varepsilon^{(t-1)}}{\partial W_{ij}} := 0$ in the Δ_{ij} adaptation-rule above.

The update-values and the weights are changed every time the whole pattern set has been presented once to the network (learning by epoch).

4. METHODS

4.1 Data Collection

Experimental data based on a study by Russell (1997) were available in electronic form. EEG data was put on an FTP site for retrieval in September 1998. The data for six individuals was then retrieved and reviewed. The data is from the STORM 97-98 laboratory studies on piloting tasks. Sample EEG data is shown in Appendix A.

Upon completion of real-time EEG data collection, a great deal of effort was put into saving the data in a compact format. The data consisted of one-second Fast Fourier Transform (FFT) samples of EEG on 16 subjects. Each subject experienced two runs for each workload over two consecutive days. Each run was slightly more than four minutes, varying only by a few seconds. Although eight channels of data representing the eight

electrodes were stored, only six EEG channels of data were further analyzed in this experiment. The utilized data were compactly stored in 12 files. Each file represented an aircraft trail distinguished by the bi-modal (forward/reverse) of the simulated aircraft and the associated velocity.

Upon reading the FFT data, plots were generated for visual perception. The plots included six channels of traditional EEG data plus two additional nodes placed close to the eyes. For each channel the FFT data was recorded in one (1) second intervals. The corresponding plots of the firs one-second interval are shown in Appendix B.

4.2 Redline Equipment Description

The Work Assessment Monitor (WAM) is a stand-alone physiological monitoring unit that was used to record the EEG data. It was interfaced with the STORM computer via a serial interface connection.

The simulator is a cockpit adapted from an F-16 Air intercept trainer whereas the subject is engaged in a simulation task of ground attach mission, using the aircraft's cannon to shoot the land target. The subject is presented with four visual displays along with auditory information, interacting with the system via a right-hand force stick and a left-hand throttle. Using the WAM, physiological data was simultaneously collected.

As the velocity of the aircraft intensified, the pilot's difficulty to accomplish the task increases accordingly. Thus, the workload was simulated by the manipulation of the velocity. Trials were used to create conditions ranging from a comfortable level of mental workload (215 knots) to an obvious overload condition (600) knots). A total of six levels of velocities (215, 325, 380, 435, 490, 600 knots) were used to simulate the workload environment. Each velocity level is assumed to simulate workload in a progressive perception scale increasing from very low workload to overload.

4.3 Preprocessing Data for Feature Selection

4.3.1 EEG Bands

A normal practice in representation of EEG data is the segmentation into frequency bands. This allows the data to be analyzed by its frequency content so as to depict which band of frequency play a vital role in EEG analysis. Using a bank of ideal digital band pass filters, these bands were extracted from the given data. Table 2 below depicts the range of frequencies used in this experiment.

Table 2. EEG Frequency Bands

Frequency Band	Frequency Rang (Hz)
Delta	DC-3
Theta	4-7
Alpha	8-12
Beta	13-30

Data from each electrode site was filtered to produce four bands of EEG, namely delta (DC-3Hz), theta (4-7Hz), alpha (8-12Hz), beta (13-30Hz). Initially, the ultrabeta band (31-42Hz) was examined, but on further investigation it was determined that the amplitudes were very small in comparison to the other bands. Thus the ultrabeta band was discarded from the set of input features.

The delta frequency band (0.1-3Hz) is usually broad or diffused, may be bilateral or widespread. It captures the human state of poor attention and low-level of arousal. The theta band is usually regional, lateralized or diffused. It is physiologically correlated with distraction or task-mind integration (Jung & Makeig, 1994; Lowe, 1998; Masic, Pfurtscheller & Flotzinger, 1995). The alpha band correlates to relaxation and composure and has a strong relationship with the occipital with eyes closed (Steuer, Shack, Grieszbach & Krause, 19XX). The beta band has a wide range and represents anything above the alpha threshold. High beta bands correlates with mental activities with high attention requirement. Mid-range beta (15-18Hz) captures state of alertness and self awareness of the immediate surround. Low beta (above 12Hz) correlates with inhibited motion; for example, a confined workspace has the tendency to increase low beta band.

4.3.2 Data Transform

The EEG has four bands and six channels each. These result in a total of 24 main features as the input to be used to the neural network. The data was then analyzed according to log power function. According to the Parseval's theorem, the power for a signal in the time domain can be calculated using its frequency representation. The following equation of log power was used:

$$p(W) = 10x\log_{10}\|F(W)\|^2 \qquad (18)$$

where

$W = frequency$

$F(W) = Fourier\ transform\ of\ the\ signal$

$p(W) = \mathbf{log}\ power\ in\ db\ scale$

Wu, Infeachor, Allen, Wilmalaratna, and Hudson (1997) have suggested an alternative to the model in equation (18).

$$P(f) = 2\Delta t\sigma^2 \qquad (19a)$$

$$A = \left|1 + \sum_{k=1}^{p} a_k \exp(-i2\lambda fk\Delta t)^2\right| \qquad (19b)$$

$$for\ \ -\tfrac{1}{2} \le f \le \tfrac{1}{2}$$

where Δt is the sampling internal, f is the normalized frequency and σ^2 is the total prediction error and can be obtained while computing α_k.

A MATLAB program was created to read the data files. As stated there was slightly less than four minutes of data per trial or per workload, resulting in about 200 seconds of data. Therefore, the neural network input consisted of approximately 200 x 24 elements of signal matrix (that is, 200 seconds of data per experimental trial x 24 trials). Each row vector with 200 data was used per workload trial, giving a total of 1200 (200 x 6) workload input vectors per subject.

4.3.3 Data Sequentation

The log power transformed EEG data was partitioned into ten seconds, i.e. 10 one-second segment at a time. To create continuity of sampling, the data was then subjected to an 80% overlap, as shown in Figure 2.

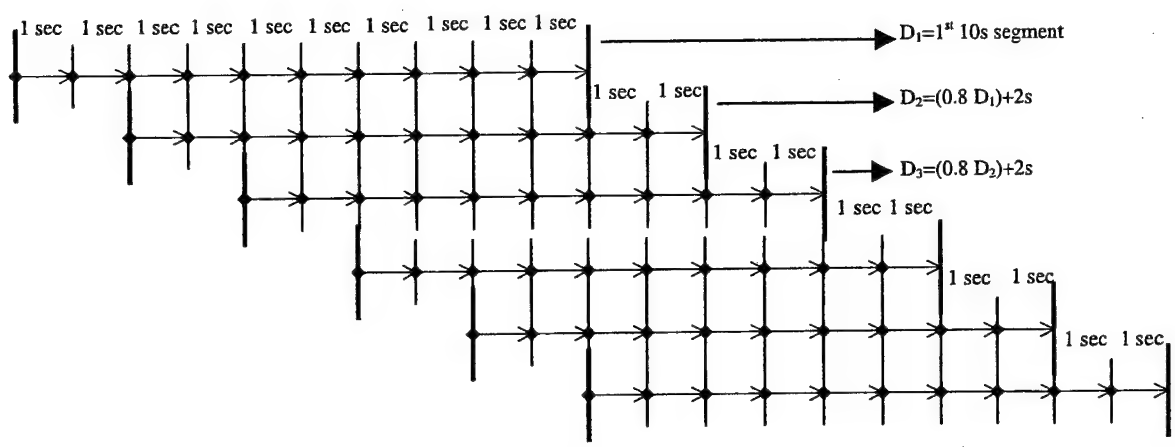


Figure 2. An illustration of segmentation procedure

In general, the procedure for the data segment is given by

$$Val = \begin{cases} \theta^{n-1} D_1 + \tau(1+\theta)^{n-2}, & n \geq 2 \\ D_1 \quad , & \tau = 0 \end{cases}$$

where Val is the data sampled, D_1 is the first t second segment; θ is the % of overlap desired, and τ is the sampling interval (in our case τ = 2 sec.).

5. NEURAL NETWORK SIMULATION EXPERIMENTS

5.1 Experimental Design

In this experiment, a four-layer, feedforward backprogpagation neural network was used. For stabilization, adaptive learning (learning rate = 0.4) and momentum (0.95) were used to decrease the time required for training the networks. The network architecture

consisted of three layers: an input layer, two hidden layers and an output layer. The input layer consisted of 24 neurons, representing features from the six channels as described earlier. The first hidden layer consisted of 18 neurons, the second hidden layer 12, and the output layer consisted of 6 neurons representing six different workloads (velocities). In this configuration, the network was fully connected with a bias input for each layer. The weight and biases were initialized using Nguyer-Widrow random generator for logistic sigmoid neurons. This technique was used to speed up the training process.

The data was divided into two groups, training and testing. Approximately 80 percent of the data were used for training the neural network. To evaluate the network's ability to separate workload initials, the data was trained to separate one class at a time, per subject. The exemplars for each experiment were randomly chosen.

The experimental design has 2 x 3 x 6 x 12 within subject analysis. There were two types of neural networks (CBNN and RBNN), three types of piloting tasks (forward, reverse, and dual), six levels of aircraft velocity (215, 325, 380, 435, 490, 600 knots), and 12 subjects.

5.2 Piloting Data Analysis

To recognize the effect of the flight task direction on the cognitive state of the pilot, a three-fold neural network system is proposed. The first network was used to perform classification on data from the forward direction flight task data. A separate network was used to perform classification on the reverse direction flight task data. Finally, a dual direction network, which combined data from both directions, was created. For the single-direction analysis, per trial input to the neural network consisted of approximately 95 vectors of length 30. This resulted in a total of approximately 570 input vectors per subject per direction. For dual-direction training, the number of input vectors approximately doubled. For applications involving automatic EEG analyses, the informative value of the available features cannot be individually determined in advance.

5.3 Selecting Salient Data for the Neural Network

The Ruck saliency measure (Ruck, et al, 1990) was used to determine which features should be used as inputs to the network. This technique calculates the partial derivative of each layer and rank orders the features based on the saliency measure. Random noise is introduced to the network as an additional input. Noise provides no information to the network, therefore, those features, which have a saliency measure lower than the saliency score of the random noise can be removed from the system. Those features removed also provide no information to the network. The equations for the Ruck saliency is:

$$\Gamma_i = \sum_p \sum_T \left| \frac{\partial z_t}{\partial x_i} \right| \qquad (21)$$

where Γ_i is the saliency for the ith feature, T is the number of outputs, p is the number of feature vectors in the training set and $\left|\frac{\partial z_t}{\partial x_i}\right|$ is the derivative of each output with respect to each input.

Saliency was computed for several preliminary runs across all subjects. The results from these preliminary runs suggest variance as a poor measure for mental workload classification in this study. Therefore, all variance features were removed since their saliency fell below the saliency of the random noise. This reduces the feature set to 85. The computer code to implement the saliency algorithm was developed by Russel (1999).

5.4 Workload Classification Rule

The output of the neural network classifier was normalized into six groups corresponding to the aircraft speeds. The linguistic vector C can represent the workload class:

C = (very low, low, medium, high, very high, unacceptable)

For the neural network to recognize the value of C for any work condition, the input data was normalized between 0 and 1 while maintaining the relationship between exemplars. Thus, C is converted from a 1 x 6 linguistic variable vector to a 6 x 6 numerical matrix. The element of C matrix is defined by:

$$P_n(i) = \frac{P(i) - P_{min}}{(P_{max} - P_{min})} \tag{22}$$

The target matrix for workload 1 through workload 6 is described below:

$$\begin{bmatrix} 0.9 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.9 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.9 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.9 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.9 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.9 \end{bmatrix}$$

Each row of the matrix represents the desired or target outputs for the six output nodes. The error goal for training was set to 0.02 (or two percent).

0

6. SAMPLE SIMULATION RESULTS

6.1 Classification by Workload (Flight Task Complexity)

The neural network algorithms (CBNN and RBNN) were applied to the data obtained from twelve pilot samples. Tables 3 and 4 give the classification results by workload complexity (aircraft control speed).

Table 3: CBNN Workload Classification by Flight Complexity

Workload Class	1	2	3	4	5	6	Mean	Std
Forward	0.8704	0.8652	0.7954	0.8646	0.8014	0.8437	0.8401	0.0336
Reverse	0.8775	0.8715	0.8316	0.7977	0.7761	0.8373	0.8319	0.0399
Dual	0.8288	0.7991	0.7696	0.7672	0.7877	0.8016	0.7923	0.0229

Table 4: RBNN Workload Classification by Flight Complexity

Workload Class	1	2	3	4	5	6	Mean	Std
Forward	0.8628	0.8379	0.7868	0.8305	0.7865	0.8358	0.8234	0.0306
Reverse	0.8636	0.8451	0.8158	0.7904	0.7581	0.8203	0.8156	0.0378
Dual	0.8200	0.7791	0.7605	0.7638	0.7688	0.8074	0.7833	0.0247

Data in Tables 3 and 4 are shown as bar charts in Figure 3.

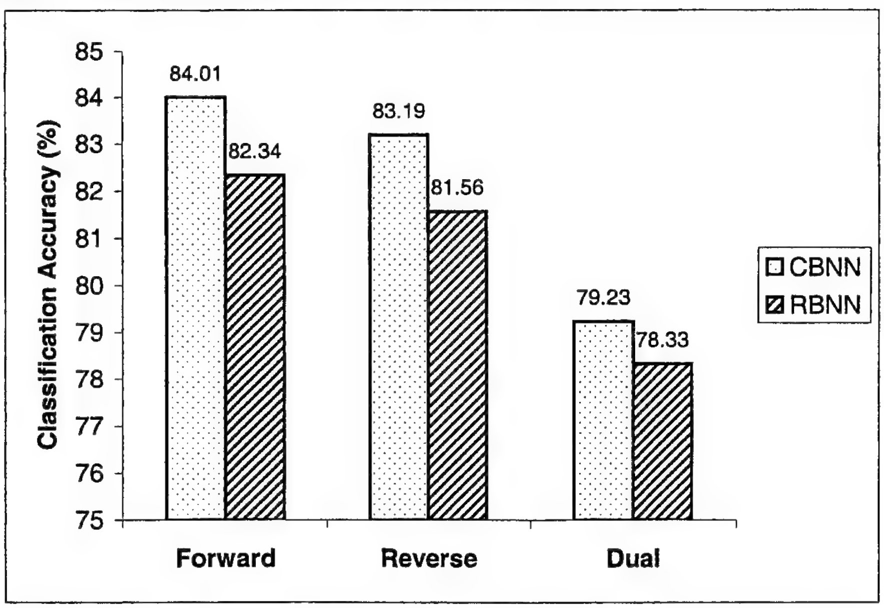


Figure 3. Comparison of mean workload classification accuracy by workloads for classical (CBNN) and resilient (RBNN) neural network models

The results show that the classical backpropagation classifier performs marginally better than the resilient backpropagation by an average difference of 1.4%. For forward flight

task, the average classification accuracies were 84.01% by CBNN and 82.34% by RBNN; for reverse flight task, the average classification accuracies were 83.19% by CBNN and 81.56% by RBNN; and for dual task, average classification accuracies were 79.23% by CBNN and 78.33% by RBNN. It should be noted that the margin of classification variability (measured by standard deviation) was high in reverse task under CBNN.

6.2 Classification Accuracy Across Subjects

Tables 5-6 give the classification results by each subject selected for analysis.

Table 5: CBNN workload classification by subject

Flight task	Subject	1	2	3	4	5	6	7	8	9	10	11	12	Mean	Std
Forward		0.85	0.86	0.85	0.86	0.79	0.88	0.77	0.83	0.84	0.80	0.88	0.87	0.8414	0.0359
Reverse		0.82	0.84	0.89	0.87	0.83	0.89	.084	0.78	0.88	0.77	0.77	0.87	0.8383	0.0442
Dual		0.78	0.79	0.87	0.80	0.72	0.85	0.75	0.76	0.84	0.74	0.80	0.81	0.7929	0.0455

Table 6: RBNN workload classification by subject

Flight task	Subject	1	2	3	4	5	6	7	8	9	10	11	12	Mean	Std
Forward		0.84	0.84	0.86	0.81	0.75	0.89	0.63	0.81	0.85	0.77	0.88	0.86	0.8246	0.0487
Reverse		0.80	0.83	0.88	0.85	0.81	0.88	.082	0.78	0.86	0.75	0.75	0.85	0.8220	0.0466
Dual		0.77	0.79	0.85	0.79	0.72	0.85	0.72	0.77	0.82	0.73	0.79	0.80	0.7838	0.0439

Data in Tables 5-6 are portrayed with bar charts in Figure 4. The results also show the 1.4% marginal classification performance of CBNN over RBNN. For forward flight task, the average classification accuracies were 84.14% by CBNN and 82.46% by RBNN; for reverse flight task, the average classification accuracies were 83.83% by CBNN and 82.20% by RBNN; and, for dual task, the average classification accuracies were 79.29% by CBNN and 78.38% by RBNN.

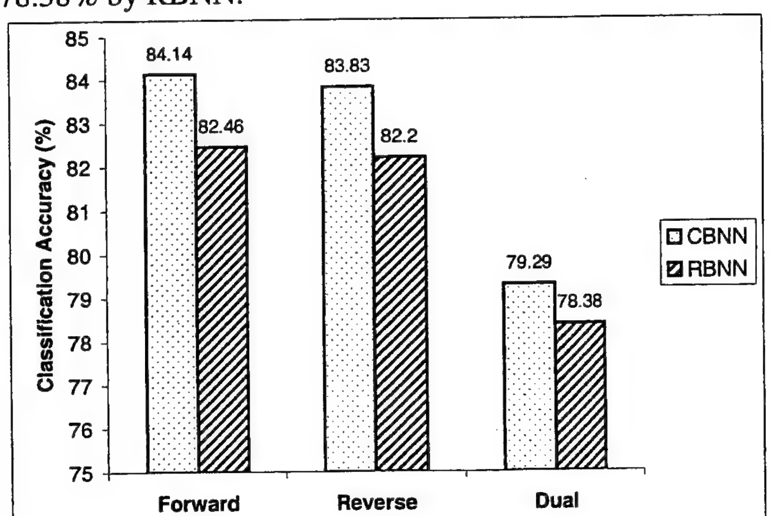


Figure 4. Comparison of mean workload classification accuracy by subjects for classical (CBNN) and resilient (RBNN) neural network models

6.3 Analysis of Processing Time Performance by CBNN and RBNN

Data were collected by the neural network simulation to attain convergence at the selected error of 2%. The number of steps are known as epochs and correlated with computation times and cost. The results are shown in Tables 7 and 8 for across subject classification times.

Table 7: CBNN Epochs Used for Result Convergence

Flight task	Subject	1	2	3	4	5	6	7	8	9	10	11	12	Mean	Std
Forward		453	264	288	364	361	218	840	249	285	498	279	331	369	169
Reverse		537	307	286	256	310	228	536	353	320	711	656	344	404	163
Dual		1329	520	665	658	733	419	1332	564	530	841	770	598	749	297

Table 8: RBNN Epochs Used for Result Convergence

Flight task	Subject	1	2	3	4	5	6	7	8	9	10	11	12	Mean	Std
Forward		209	81	137	167	177	45	700	97	107	253	73	160	184	173
Reverse		346	130	111	74	129	64	337	217	109	528	613	139	233	183
Dual		294	123	156	163	153	93	315	133	160	222	152	157	177	67

The average number of epochs in Tables 7 and 8 are shown in Figure 5. The results show that RBNN takes shorter number of iterations to achieve the desired results. The CBNN model takes more iterations, about 50% above RBNN. A high number of iterations (749) was observed during dual task processing by CBNN compared to only 177 iterations on the same task by RBNN.

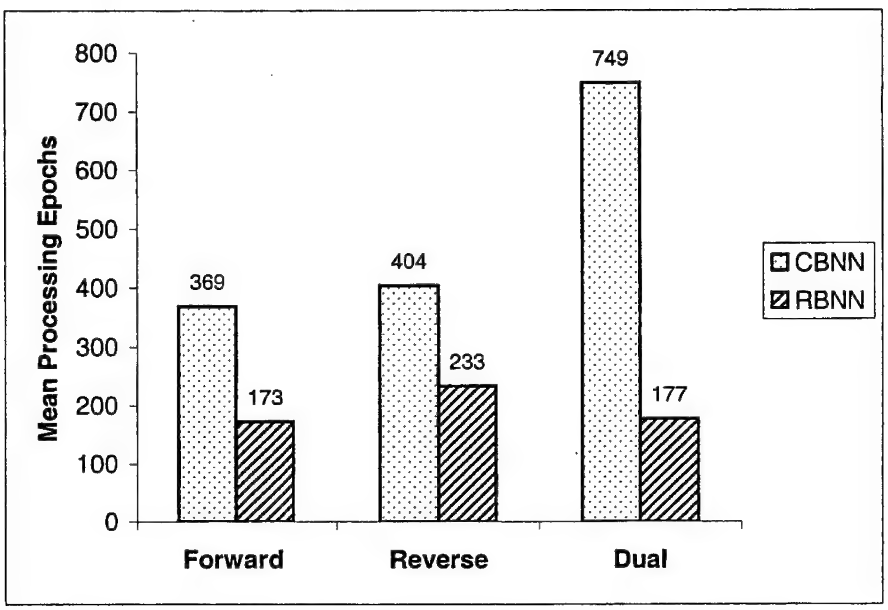


Figure 5. Comparison of mean processing epochs for classical (CBNN) and resilient (RBNN) neural network models

In addition to the number of iterations, we also observe the behavior of the learning curves as exhibited in Appendix C. The results show that the RBNN produces smooth

and gradual learning curves characterized by multiple local minima (valleys). One main reason for this smooth and gradual learning curve by the RBNN is that it performs a local adaptation of the weight-updates according to the behavior of the error function and is not blurred by the unforeseeable influence of the size of the derivative but only dependent on the temporal behavior of its sign (Riedmiller & Braum, 1993).

6.4 User Guides to Software Applications

Two types of programs have been developed for this project, the neural network build-in functions, and the application programs. All the programming are written with **Matlab** soft package. Their names and uses are described as follows:

Neural Network Built-in Functions

plotperf.m	Function called by neural network toolbox, to plot network performance
traingdx.m	Function called by neural network toolbox to use Gradient descent with momentum & adaptive learning backpropagation paradigm to train neural network
trainrp.m	Function called by neural network toolbox to use resilient backpropagation paradigm to train neural network

Application Programs

Load_FFT.m	Pre-process raw FFT data;
Classify_CBNN.m	Main file to classify workload using CBNN model;
Classify_RBNN.m	Main file to classify workload using RBNN model;
LOAD_AFBEEG.m	Sub-program of **Classify_CBNN.m** and **Classify_RBNN.m**, to prepare input and output matrix data for neural networks.
Val_afb.m	Sub-program of **Classify_CBNN.m** and **Classify_RBNN.m**, to construct confusion matrix

Three built-in functions in **Matlab Neural Network toolbox** have been modified. Before running the application programs, these functions need to be installed in the **Matlab Neural Network toolbox**.

The application programs need to be installed in a suitable directory as well. When running the application programs, the file **Load_FFT.m** must be run first to pre-process the raw FFT data. Since the files: **LOAD_AFBEEG.m** and **Val_afb.m** are sub-programs called by main program, they are not going to be run by themselves.

The user's guide of installing all these files and running some application forms are described below one by one.

1. **User Guide to install Neural Network Built-in functions**

The procedures to install the built-in functions are introduced as follows:

1) Open the **File Manager** of the computer;
2) Locate the directory where **Matlab** software is saved;
3) Locate the sub-directory of '**nnet**' following '**../Matlab/Toolbox/nnet/nnet**';
4) Rename the built-in program '**plotperf.m**' as '**plotperf_old.m**';
5) Save the modified **plotperf.m** file to the **nnet** directory

2. User Guide to install application files to directories

The procedures to install application files in place are introduced as follows:

1) Open the **File Manager** of the computer;
2) Establish a sub-directory under **C:** drive as **C:/EEG;**
3) Save all the application files under directory **C:/EEG**

3. User Guide to run Load_FFT.m file

Before running the program, make sure your unzipped FFT data are under the sub-directory **C:/EEG/datafft**. Also, make a directory of **C:/EEG/dataeeg** to hold the data prepared by program **Load_FFT.m**.

1) Run **Matlab** Soft package;
2) After suggestion sign '>>', type '**cd C:/EEG**' to enter the sub-directory **EEG;**
3) After suggestion sign '>>', type '**Load_FFT**' to run the file to load FFT data and manipulate it into the format fit for input into neural network;
4) When prompted with '**enter the desired segment size in seconds:**', enter '**10**' because we are using the ten seconds time window;
5) Then you will be prompted **with 'enter the amount of desired overlap in whole %:**', enter '**80**' now because here we prepared the data with 80% overlap as input to neural network;
6) The program with load all the FFT data to the desired format and save them to a sub-directory **C:/EEG/dataeeg**

4. User Guide to run CBNN Model

Before running the program, make a directory of **C:/EEG/RESULTS** to hold the file containing the running outputs of the program **Classify_CBNN.m**.

The procedures to run the **CBNN** Model are described as follows:

1) Run **Matlab** Soft package;
2) After suggestion sign '>>', type '**cd C:/EEG**' enter the sub-directory **EEG**;
3) After suggestion sign '>>', type '**Classify_CBNN**' to run the Classical Backpropagation Neural Network (**CBNN**) file;
4) The program will prompted you with '**Choose subjects by number (1-12) Numerical Codes of SUBJECTS to Examine []:**', you have two choices here,

a. If you want to run **one** subject, just enter one number from 1 to 12, for example, type '**3**', to run the third subject, which is subject '**bk**';
b. If you want to run **more than one** subject, enter the numbers with a bracket, for example, type **'[3:5]'** to run subject **3, 4,5**, or you can type **'[3;5]'** to run only subject **3** and **5**;

5) Then comes another prompt '**Choose the directions: (1-3) 1(Forward), 2(Reverse) and/or 3(Dual) []:**', again you have two choices,
   a. If you want to run only **one direction**, you can type in one number **from 1 to 3**. For example, type '**1**', to run the **Forward direction**;
   b. If you want to run **more than one direction**, enter the numbers with a bracket, for example, type **'[1:3]'**, to run **three directions**, or '**[1;3]**' to run **two directions, forward and dual**.
6) The third prompt is '**Choose the number of total runs (<11) per subject per direction, For each direction, each subject will be run x times. x=**'. Here, type in one number **from 1 to 10**, for example, type '**5**', to run the neural network **5** times.

5. User Guide to run RBNN Model

Before running the program, make sure a directory of **C:/EEG/RESULTS** exist, to hold the file containing the running outputs of the program **Classify_RBNN.m**.

The procedures to run the **RBNN** Model are described as follows:

1) Run **Matlab** Soft package;
2) After suggestion sign '>>', type '**cd C:/EEG**' enter the sub-directory **EEG**;
3) After suggestion sign '>>', type '**Classify_RBNN**' to run the Resilient Backpropagation Neural Network (**CBNN**) file;
4) The program will prompted you with '**Choose subjects by number (1-12) Numerical Codes of SUBJECTS to Examine []:**', you have two choices here,
   a. If you want to run **one** subject, just enter one number from 1 to 12, for example, type '**3**', to run the third subject, which is subject '**bk**';
   b. If you want to run **more than one** subject, enter the numbers with a bracket, for example, type **'[3:5]'** to run subject **3, 4,5**, or you can type **'[3;5]'** to run only subject **3** and **5**;
5) Then comes another prompt '**Choose the directions: (1-3) 1(Forward), 2(Reverse) and/or 3(Dual) []:**', again you have two choices,
   a. If you want to run only **one direction**, you can type in one number **from 1 to 3**. For example, type '**1**', to run the **Forward direction**;
   b. If you want to run **more than one direction**, enter the numbers with a bracket, for example, type **'[1:3]'**, to run **three directions**, or '[1;3]' to run **two directions, forward and dual**.
6) The third prompt is '**Choose the number of total runs (<11) per subject per direction, For each direction, each subject will be run x times. x=**'. Here,

type in one number **from 1 to 10**, for example, type '**5**', to run the neural network **5** times.

7. DISCUSSION AND CONCLUSION

7.1 Discussion

Our study shows that the overall average classification accuracy were consistent with 1.4% margin of differences between CBNN and RBNN. However, if processing time and cost are to be minimized, the RBNN is recommended. We also observed that by using either the CBNN or RBNN, the classification accuracy of dual task performance across subjects and flight complexity lags behind forward and reverse flight tasks by a margin between 3-6%. These observed classification differences can be attributed to the perceptual and cognitive complexity involved with dual task processing (Wickens, 1984). The result further illustrates that the neural network models develop confusions in differentiating task modalities: forward and backward synchronicity versus tasks processed independently.

Other observations are worth discussing. As shown in Tables 3 and 4, for forward flight task, the classifications accuracies by CBNN were over 80% for all flight speeds except speed level #3 (380 knots) that was less than 80%, and was 78.65% classification by RBNN at 490 knots. The highest classification accuracy was for subject #6 (Tables 5 & 6), with scores as high as 89% for reverse task (by CBNN) and forward task (by RBNN). The worst classification was for subject #7 with 63% classification on forward task (by RBNN) and 75% classification on dual task (by CBNN).

Based on the sample simulation experiments, we also collected data on misclassified workload for each of the task categories. Tables 9 - 10 give examples for forward flight task by the CBNN and RBNN.

Table 9: Sample Workload Classification Table for Forward Flight Task Using CBNN

Exemplar Data Class	% Classification by CBNN						Total % Misclassification
	1	2	3	4	5	6	
1	87.1	4	2.6	1	0	0	6.6
2	5	86.7	3	1	1	0	10.
3	1.5	2.6	79.6	2.3	0	0	6.4
4	0	1.8	3.3	86.5	4.3	0	9.4
5	0	0.8	2.4	3.9	80.1	3.1	10.2
6	0	0	1.7	2.3	4.9	84.5	8.9

Table 10: Sample Workload Classification Table for Forward Flight Task Using RBNN

Exemplar Data Class	% Classification by CBNN						Total % Misclassification
	1	2	3	4	5	6	
1	86.51	3.2	2.9	1.2	0	0	7.3
2	2.6	84.27	4.3	0.75	0.45	0	8.1
3	1.0	3.6	78.71	2.0	0	0	6.6
4	0	0.8	1	83.15	4.3	1.8	7.9
5	0	0	1.2	2.0	78.51	2.8	6.0
6	0	0	0.3	4.8	3.4	82.99	8.5

The meaning of data in Tables 9-10 is as follows: when a known workload class exemplar is presented to the neural network, the percentage classification accuracy is observed. For example, in Table 9, the entry on row 1, column 1, is 87.1%. This means that the CBNN accurately classify workload #1 (very low workload) 87.1% of the time during the experiment. Other entries corresponding to row 1 indicates percentage misclassification. For example, the value of 4% on row 1, column 2, indicates that the network misclassify very low workload (speed level 215 knots) as low workload (speed level, 325 knots). The last column in Tables 9-10 indicate the total percentage misclassification.

Note that the misclassification data vary according to the simulation experiments. Similar data on Tables 9-10 can be obtained for reverse and dual flight tasks. Our experiments show that on the average, the percentage of misclassification across task context was an average of 4.5% less than the expected theoretical value of 16.6% (i.e. 1 chance in 6).

7.2 Conclusion

The results obtained demonstrate the ability to classify human workload levels by the use of backpropagation neural networks. The results show the ability to differentiate human perception of workload through data generated by evoked potentials from EEG measures. There is little doubt that, if appropriately used artificial neural networks offer a robust method for analyzing human signals related to work performance. The most challenging task is selecting data pre-processors and feature selection methods for the neural network models.

We can also argue that classification of EEG signal data by neural network also provide a description of human perception and control of dynamic actions, both in time and space. In fact, by analyzing individual human signals, it is possible to determine the correlation between performance at each frequency bands (delta, theta, alpha, beta) and the level of training needs, or, even the modality of task processing that best fits the human operator. This assertion fits well with task types: attention, motor, or cognitive.

By examining the results of the experiments, we observed that the variations in classification results are also dependent on the ability of the subjects and task complexity.

Future experimental analysis should look at the effects of skill differentiation on the classifier algorithms. Comparisons between skill levels will be a challenging modeling problem, but the results obtained may help to elucidate how people perform at the level of overload task processing.

We like to offer recommendations for future research directions:

1) The results obtained in this report may further be investigated with a recurrent neural network (Hazarika, Tsoi & Sergejeio, 1997). A recurrent neural network is similar in nature to the multi-layer perception, except that it may include feedback and time delays. This can provide a comparative basis for training performance with and without feedback.
2) Extend the human signal data to include other physiological measures such as heart rate, respiration rate, and / or eye blink activity (Russell and Wilson, 1996).
3) Investigate the availability of other feature selection algorithms that may perform better than Ruck saliency measures. For example, the use of factor loading or discrimination analysis across frequency bands and task types.
4) Data on the pilot's subjective rating of each task should be incorporated into the EEG analysis. The classification can then be done with the hybrid model of fuzzy set theory and neural network.

REFERENCES

Bird, B. L., Newton, F. A., et al (1978). Behavioral and electroencephalographic correlates of 40-Hz EEG biofeedback training in humans. Biofeedback and Self-Regulation, 3, 13-28.

Fruhstorfer, H., and Bergstrom, R. (1969). Human vigilance and auditory evoked responses. Electroencephalography and Clinical Neurophysiology, 27(4), 346-355.

Fukuda, O., Tsuji, T., and Kaneko, M. (1995). Pattern classification of EEG signals using a log-linearized Gausssian mixture neural network. IEEE International Conference on Neural Networks (2479-2484). Piscataway, NJ: IEEE Press.

Fukuda, O., Tsuji, T., and Kaneko, M. (1996). Pattern classification of time-series EEG signals using neural networks. Procedings of IEEE International Workshops on Robot and Human Communication (217-222). Piscataway, NJ: IEEE Press.

Gevins, A and Morgan, N. H. (1988). Application of neural network (NN) signal processing in brain research. IEEE Trans. On Acoustics, Speech, & Signal Processing, Vol. 36, pp. 1152-1161.

Gevins, A. Zeitlin, G., Yingling, J., Doyle, J., Dedon, M., Schaffer, R., Roumasset, J., and Yeager, C. (1979). EEG patterns during cognitive tasks: I. Methodology, and analysis of complex behaviors. Electroencephalography and Clinical Neurophysiology, Vol., 47, pp. 693-703.

Greene, K. A., Bauer, K. W., Kabrisky, M., Rogers, S. K., Russell, C., and Wilson, G. F. (1966). A preliminary investigation of selection of EEG and psychophysiological features for classifying pilot workload. In Intelligent Systems through Artificial Neural Networks, Vol. 6, (C. H. Dagli, M. Akay, C. L. P. Chen, B. R., Fernandez, & J. Ghosh, Eds.), New York: ASME Press, pp. 691-697.

Grossberg, S. (1982). Studies of Mind and Brain. Boston, MA: D. Reidel Publishing Company.

Hancock, P. Meshkati, N., and Roberston, M. (1985). Physiological reflections of mental workload. Aviation, Space and Environmental Medicine, pp. 1110-1114.

Hazarika, N. Tsoi, A. C., and Sergejew, A. A. (1997). Nonlinear classifications in EEG signal classification. IEEE Transactions on Signal Processing, 45(4), 829-836.

Hellyar, M. T., Ifeachor, E. C., Mapps, D. J., Allen, E. M. and Hudson, N. R. (1995). An expert system approach to EEG signal processing. Knowledge-based Systems, 8(4), 164-173.

Hiraiwa, A., Simohara, K., and Tokunaga, Y. (1989). EMG pattern analysis and classification by neural network. IEEE International Conference on Systems, Man, and Cybernetics (1113-1116). Piscataway, NJ: IEEE Press.

Horst, R. and Dochin, E. (1980). Beyond averaging II, Single trial classification of exogenous event-related potentials using stepwise discriminate analysis. Electroencephalograph and Clinical Neurophysiology, Vol. 48, pp. 113-126.

Horst, R., Munson, R. and Ruchklin, D. O. (1984). Event-related potential indices of workload in a single task paradigm. Proceedings of the Human Factors Society, 28th Human Meeting, Santa Monica, CA: Human Factors Society.

Humphrey, D., and Kramer, A. F. (1993). Towards psychophysiological assessment of dynamic changes in mental workload. Technical Report UIUC-Bii-HPP-93-01. Urbana-Champaign: The University of Illinois.

Israel, J., Wickens, C., Chesney, G., and Donchin, E., (1980). The event-related brain potential as an index of display monitoring workload. Human Factors, Vol. 22, pp. 211-224.

Itoh, Y., Hayashi, Y., Tskui, I., and Saito, S. (1989). Heart rate variability and subjective mental workload I flight task validity of mental workload measurement using HRV method. In Work with Computers: Organizational, Management, Stress and Health Aspects (M. J. Smith & Salvendy, Eds.), Amsterdam: Elsevier, pp. 209-216.

Jasen, B. H., and Dawant, B. M. (1989). Knowledge-based approach to sleep EEG analysis-a feasibility study. IEEE, Transactions on Biomedical Engineering, 360, 510-518.

Jung, T.P., Makeig, S. (1994). Estimating level of alertness from EEG, Engineering in Medicine and Biology Society, Engineering Advances: New Opportunities for Biomedical Engineers. Proceedings of the 16th Annual International Conference of the IEEE, P vol.2, pp.1103-1104

Keirn, A. A., and Aunon, J. I. (1990). A new mode of communication between man and his surroundings. IEEE Transactions on Biomedial Engineering, Vol. 37(12), pp. 1209-1214.

Kirk, B.P and Lacourse, J.R. (1997), Performance monitoring from the EEG power spectrum with a radial basis function. Proceedings of the IEEE 1997 23rd Northeast neural network Bioengineering Conference, pp.19-20

Klimesh, W., Schimke, H. M. and Pfurtscheler, G. (1993). Alpha frequency, cognitive load, and memory performance. Brain Topography, Vol. 5, pp. 241-259.

Kloppel, B. (1994). Application of neural networks for EEG analysis: Considerations and first results. Neuropsychobiology, Vol. 29, pp. 39-46.

Kloppel, B. (1994). Neural networks as a new method for EEG analysis: A basic introduction. Neuropsychobiology, 29, 33-38.

Kramer, A. F. and Stryer, D. (1988). Assessing the development of automatic processing: An application of dual-task and event-related brain potential. Psychophysiology, Vol. 23, pp. 33-47.

Kramer, A. F., (1991). Physiological metrics of mental workload: A review of recent progress, In Multiple Task Performance (D. Damos, Ed.) Washington: Taylor & Francis, pp. 279-328.

LeCavaller, M., Miller, G. E., and Klemm, W. R. (1988). Analysis of evoked EEG patterns from controlled cognitive states. Proc. Of 10th Annual International Conference on IEEE Engineering in Medicine and Biology Society (1196-1197). Piscataway, NJ: IEEE Press.

Lindholm, E., Cheatham, C., Koriath, J, and Longridge, JR (1984). Physiological assessment of aircraft pilot workload in simulated landing and simulated hostile threat environments. Air Force Systems Command, Technical Report AFHRL-TR-83-49. Williams Air Force Base, Arizona.

Lowe, D. (1998), Feature space embeddings for extracting structure from single channel wake EEG using RBF networks, Proceedings of the 1998 IEEE Signal Processing

Society Workshop on Neural Networks for Signal Processing, pp.428–437.
Masic, N., Pfurtscheller, G., and Flotzinger, D. (1995). Neural networks based predictions of hand movements using simulated and real EEG data. Neurocomputing, 7, 259-274.
McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas imminent in nervous activity. Bulleting of Mathematical Biophysics, 5, 115-133.
Morton, P.E., Tumey, D.M., Ingle, D.F., Downey, C.W., Schnurer, J.H.(1991), Neural network classification of EEG data generated through use of the audio oddball paradigm, Proceedings of the 1991 IEEE Seventeenth Annual Northeast Bioengineering Conference, pp.7–8
Moser, J. and Annoy, J. (1986). Classification and detection of single evoked brain potential using time-frequency amplitude features. IEEE Transactions on Biomedical Engineering, Vol. 33, pp. 1096-1106.
Paradey, J., Roberts, S., and Tarassenk L. (1994). Applications of Application of artificial neural networks to medical signal, IEE Colloquium on Applications of Neural Networks to Signal Processing (Digest No.1994/248), pp.9/1-9/3.
Peltoranta, M., and Pfurtscheller, G. (1994). Neural network based classification of non-averaged event-related EEG responses. Medical and Biological Engineering and Computing, 32, 189-196.
Peters, Bo. O. Pfurtscheller, G., and Flyvbjerg, H. (1998). Mining multi-channel EEG for its information content: and ANN-based method for brain-computer interface. Neural Networks, 11, 1429-1433.
Reddy, D.C., Rao, K.D. (1992). Neural Networks for Classification of Multi channel EEG Signals, Engineering in Medicine and Biology Society, 1992. Vol.14. Proceedings of the Annual International Conference of the IEEE, pp.1024 –1025
Riddington, E. P., Ifeachor, E. C., Allen, E. M., Hudson, N. R., and Mapps, D. J. (1994). A fuzzy expert system for EEG interpretation. Proceedings of the International Conference of Neural Networks and Expert Systems in Medicine and Health Care, Plymouth, UK, 291-302.
Riedmiller, M. and Braun, H. (1993). A direct adaptive method for faster backpropagation learning: The RPROP Algorithm. Proceedings of the IEEE International Conference on NN (ICNN), San Francisco, pp. 586-591.
Roberts, S.J., Krkic, M., Rezek, I., Pardey, J., Tarassenko, L. Stradling, J., Jorden, C. (1995). The use of neural networks in EEG analysis, IEE Colloquium on Sleep Monitoring, pp.3/1 -3/3.
Roscoe, A. H. (1975). Heart rate monitoring of pilots during steep-gradient approaches. Aviation, Space and Environmental Medicine, Vol. 47, pp. 86-90.
Russell, C. and Wilson, G. F. (1996). Mental workload classification using a backpropagation neural network. In Intelligent Engineering System through Artificial Neural Networks. Vol. 6 (C. H. Dagli, M. Akay, C. L. P. Chen, B. R. Fernanclez and J. Ghost, Eds.) New York: ASME Press, pp. 685-695.
Shiao-Lin, L., Yi-Jean, T., and Cheng-Yuan, L. (1993). Conscious mental tasks and their EEG signals. Medical & Biological Engineering & Computing, Vol. 31, pp. 421-425.
Skelly, J., Purvis, B., and Wilson, G. (1989). Fighter pilot performance during airborne and simulator missions physiological comparisons. In Electric and Magnetic

Activity of the Central Nervous System: Research and Clinical Applications in Aerospace Medicine (K. Jensen, Ed.). France, NATO AGARD.
Sterman, B. Schuminer, G., Dushenko, T., and Smith, J. (1987). Electroencephologographic correlates of pilot performance: Simulation in-flight studies. In Electrical and Magnetic Activity of the Central Nervous System: Research and Clinical Applications in Aerospace Medicine (K. Jensen, Ed.) France: NATO AGARD.
Sterman, M. B, et. al (1992). Quantitative EEG patterns of differential in-flight workload. Space Operations Applications and Research, pp. 466-473
Steuer, D., Schack, B., Grieszbach, G., Krause, W. (1995). Classification of human cognitive processes by the use of an improved neural backpropagation network. Proceedings of IEEE International Conference on Neural Networks, vol.5, pp.2495-2500.
Sun, M., Ryan, N.D., Dahl, R. E., Hsin, H. C., Iyengar, S and Sclabassi, R.J. (1993). A neural network system for automatic classification of sleep stages. Proceedings of the Twelfth Southern Biomedical Engineering Conference, pp.137-139
Tsio, A. C., So, D. S. C., and Sergejew, S. (1994). Classification of electroencephalogram using artificial neural networks. Neural Information Processing Systems, Vol. 6, pp. 1157-1158.
Wang, Z., He, Z. and Chen, J. D. Z. (1999). Filter basics and neural network-based feature extraction and automatic classification of electrogastrogram. Annals of Biomedical Engineering, Vol. 27, pp. 88-95.
Wickens, C. D. (1984). Processing resources in attention. In R. Parasuraman & D. R. Davies (Eds.). Varieties of Attention (pp. 63-102). San Diego, CA: Academic Press.
Wierwille, W. (1979). Physiological measures of aircrew mental workload. Human Factors, Vol. 21 pp. 575-594.
Wilson, G. F. (1993). Air-to-Ground training mission: A psychophysiological workload analysis. Ergonomics, Vol., 36(9), pp. 1071-1187.
Wilson, G., and O'Donnell, R. (1986). Steady evoked responses: Correlations with human cognition. Psychophysiology, Vol. 23, pp. 57-61.
Wilson, G., and O'Donnell, R. (1988). Measurement of operator workload with neuropsychological workload test pattern. In human Mental Workload (P. Hancock and N. Meshkati, Eds.), Amsterdam: Elsevier.
Wilson, G., Purvis, B., Skelly, J. Fullerkamp, P., and Davis, I. (1987). Physiological data used to measure pilot workload in actual flight and simulator conditions. Proceedings of the Human Factors 31st Annual Meeting (pp. 161-164). New York: Human Factors Society.
Wu, J., Ifeachor, E. C., Allen, E. M. and Hudson, N. R. (1994). A neural network based artefact detection system for EEG signal processing. Proceedings of the International Conference on Neural Networks and Expert Systems in Medicine and Health Care, Plymouth, U.K, 257-266.
Wu, J., Ifeachor, E. C., Allen, E. M., Wimalaratna, S. K., and Hudson, N. R. (1997). Intelligent artefact identification in electroencephalography signal processing. IEEE Proc. Scientific Measurement Technology, 144 (5), 193-201.

Xiao, D, Yang, H., and Zhou, S. (1997). Extraction of 40-Hz EEG bursts for chaos analysis of brain function. IEEE Engineering in Medicine and Biology (July/August), 27-32.

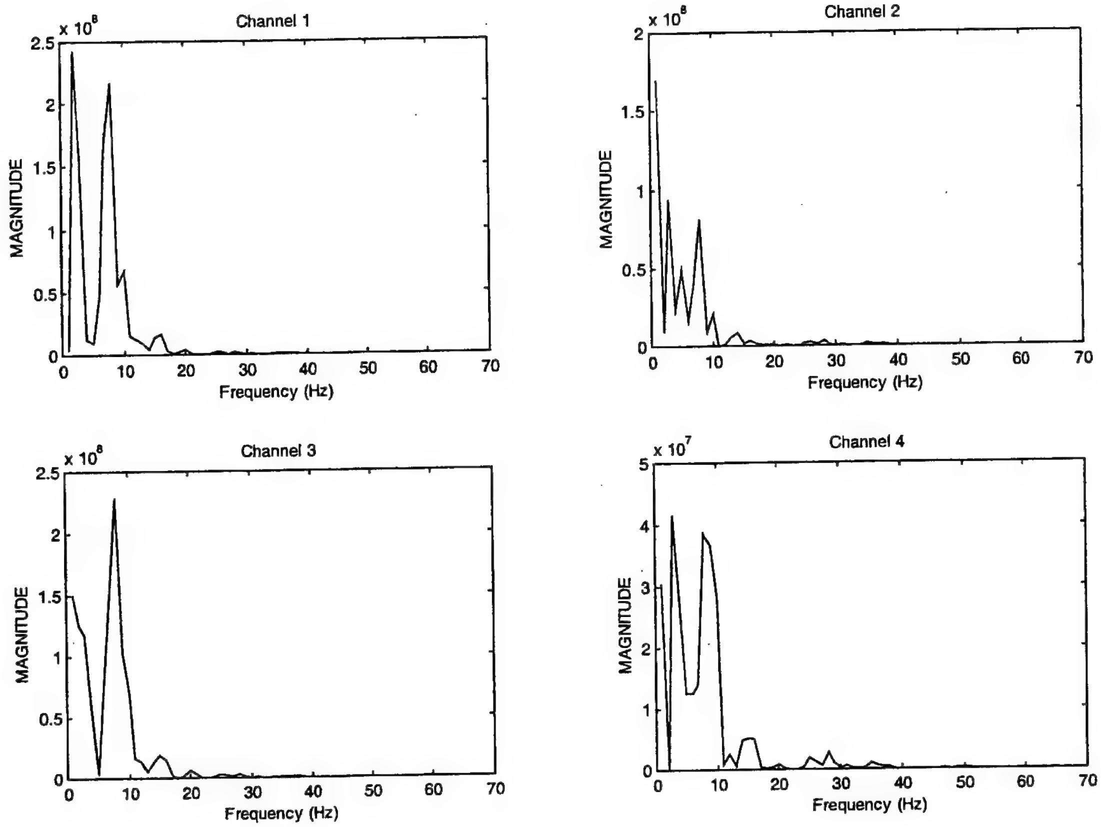
Ye, N. (1997). Neural networks approach to user modeling and intelligent interface: A review and reappraisal. International Journal of Human-Computer Interaction, 9(1), 3-23.

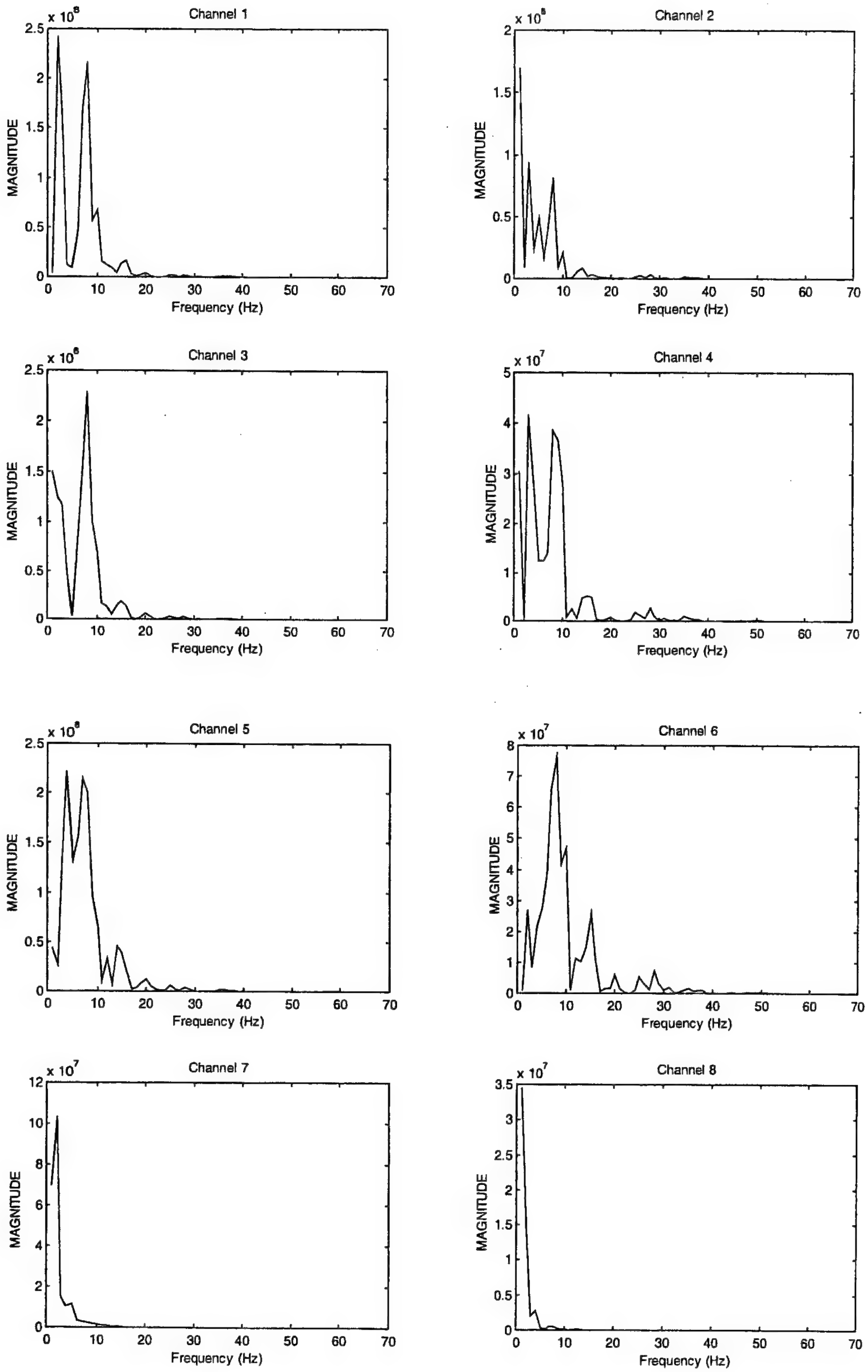
Zhu, J., Hazarika, N. Tsoi, A. C., and Sergejew, A. A. (1994). Classification of EEG signals using wavelet coefficients and an ANN. Presented at the Pan Pacific Workshop on Brain Electric and Magnetic Topography, Sydney Australia (CD-ROM).

Appendix A

Appendix B

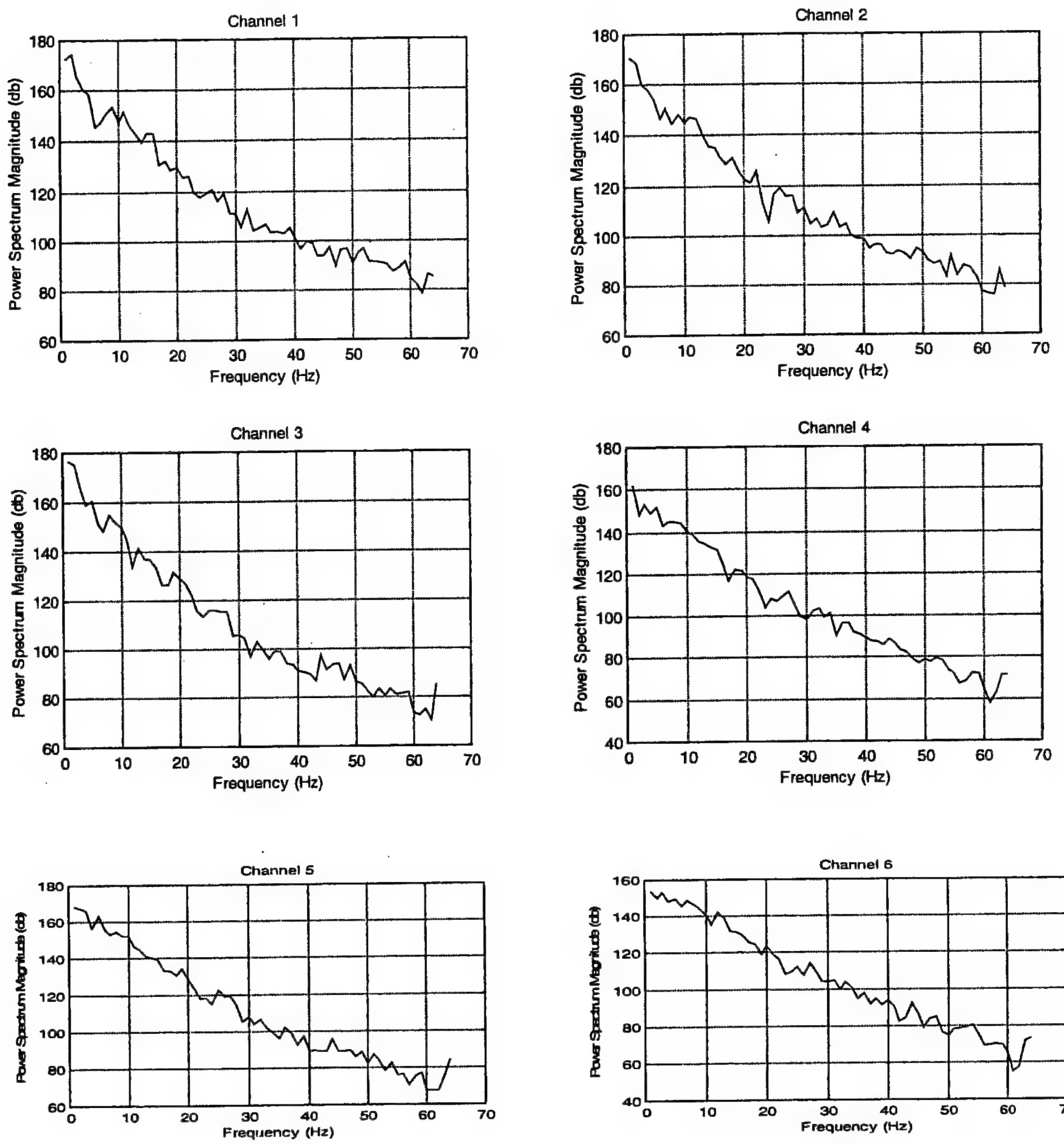
FFT MAGNITUDE PLOTS (SUBJECT:BK, 1ST SECOND INTERVAL)



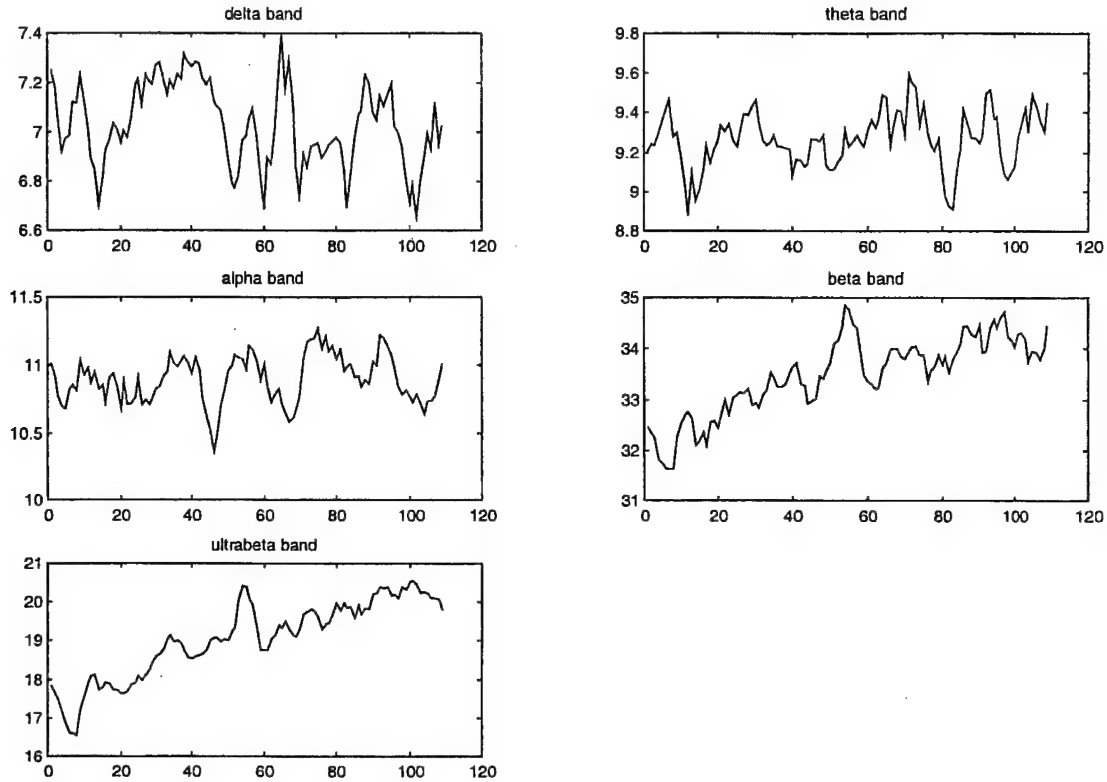


Channel 1
x 10^8
MAGNITUDE
Frequency (Hz)
Channel 2
x 10^8
MAGNITUDE
Frequency (Hz)
Channel 3
x 10^8
MAGNITUDE
Frequency (Hz)
Channel 4
x 10^7
MAGNITUDE
Frequency (Hz)
Channel 5
x 10^8
MAGNITUDE
Frequency (Hz)
Channel 6
x 10^7
MAGNITUDE
Frequency (Hz)
Channel 7
x 10^7
MAGNITUDE
Frequency (Hz)
Channel 8
x 10^7
MAGNITUDE
Frequency (Hz)

POWER SPECTRUM MAGNITUDE (db) OF SIX CHANNELS USED (SUBJECT: BK, MEAN VALUE OF FIRST 10 SECONDS SEGMENT)



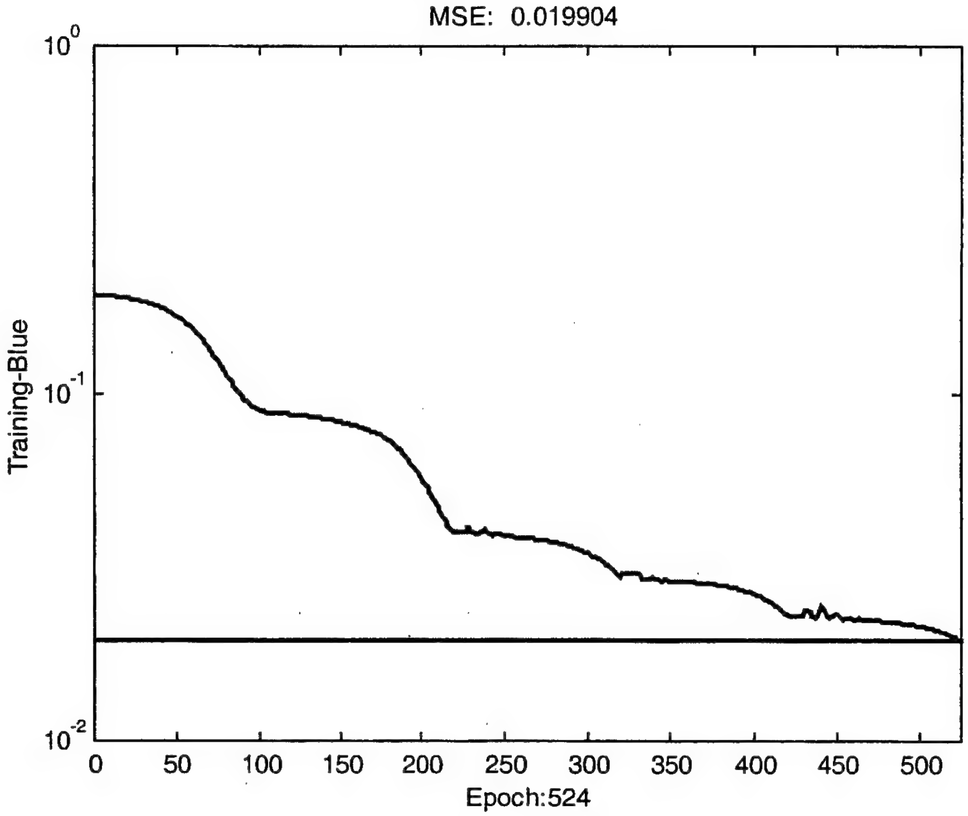
SAMPLE PLOT OF FIVE FILTERED FREQUENCY BANDS
(SUBJECT: BK, CHANNEL 3)



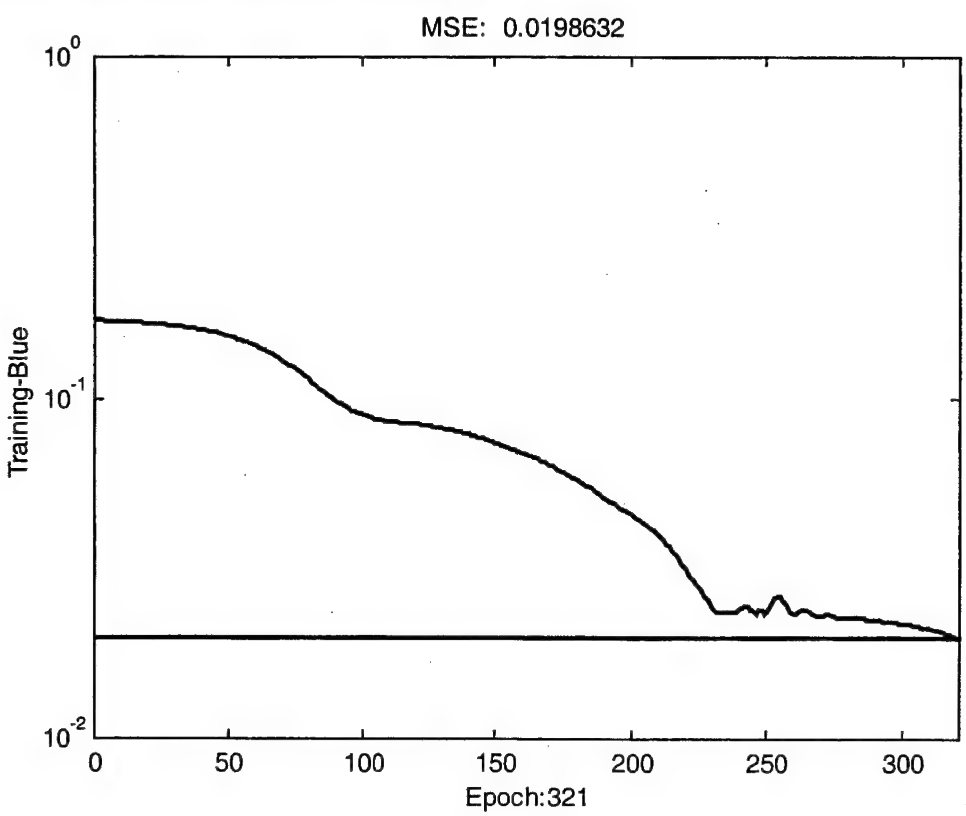
Appendix C

Sample Plots of Learning Curves for CBNN and RBNN

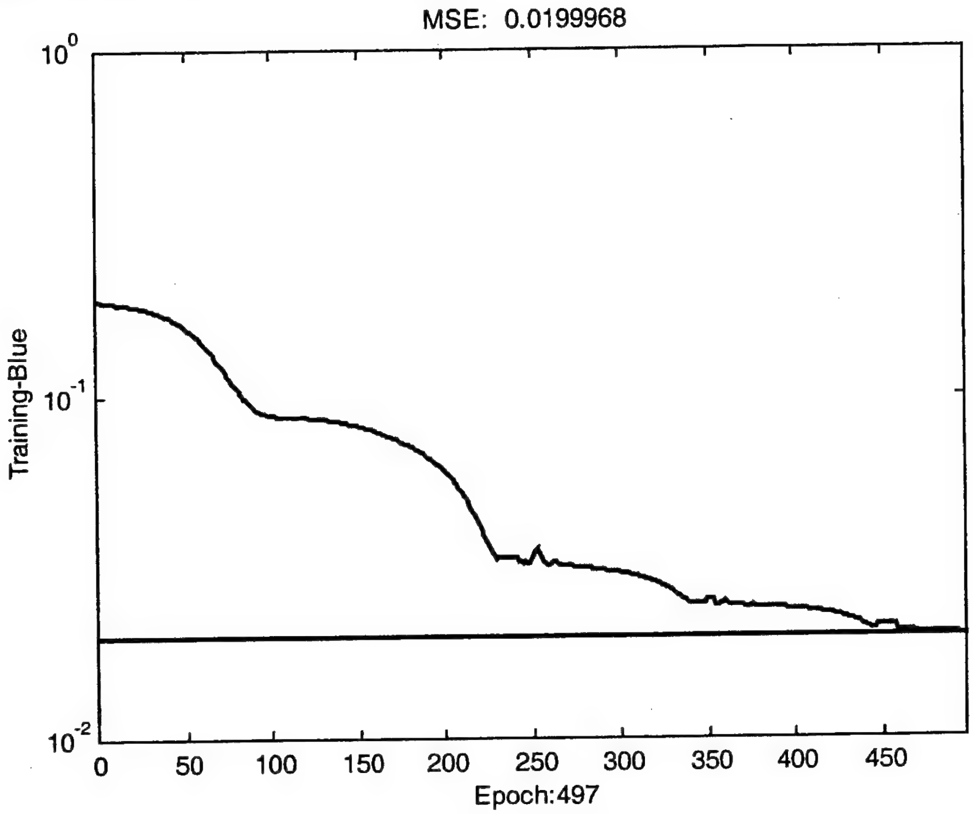
SUBJECT: AB, CBNN, FORWARD DIRECTION



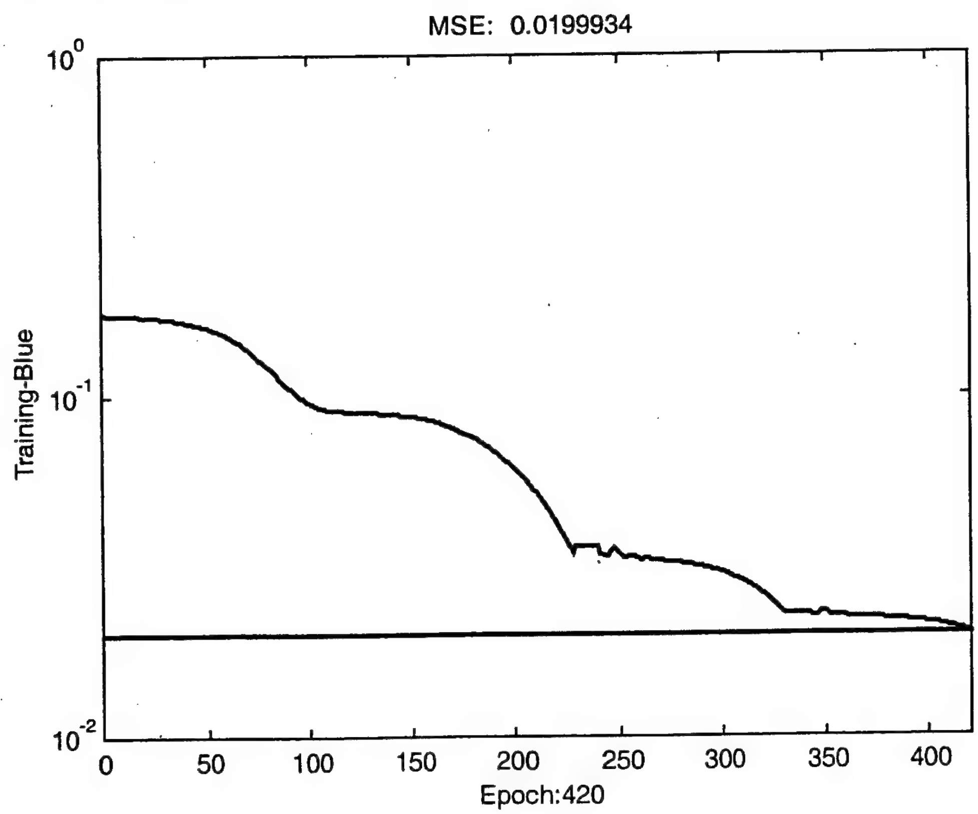
SUBJECT: BK, CBNN, FORWARD DIRECTION



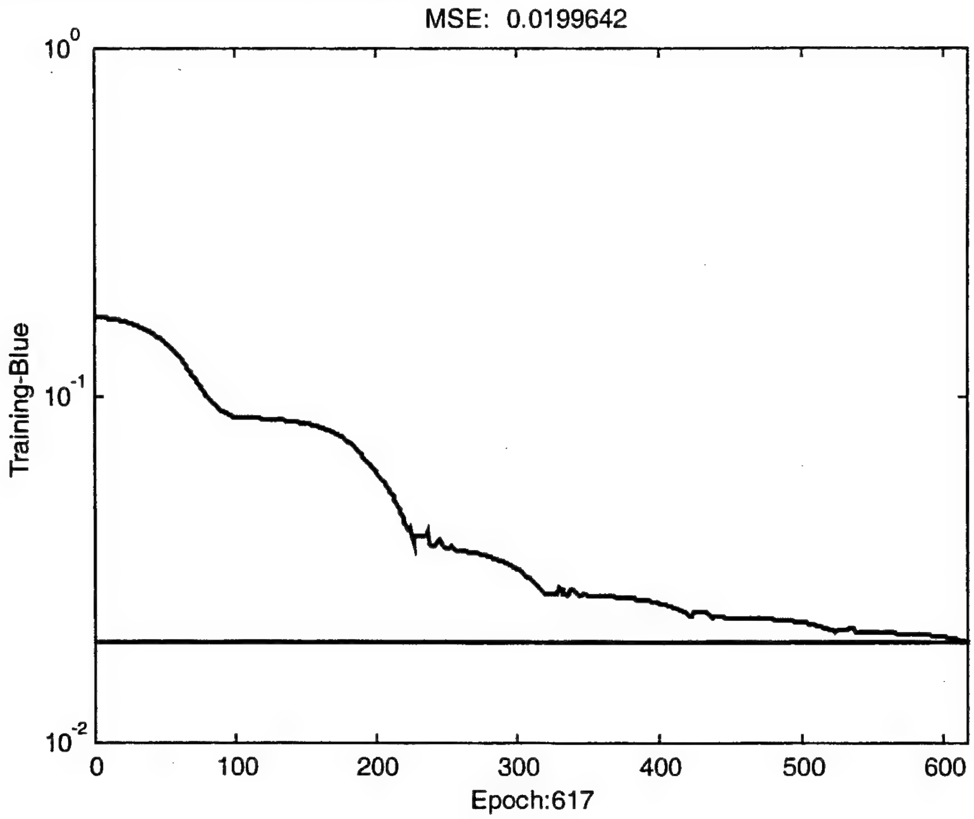
SUBJECT: ES, CBNN, FORWARD DIRECTION



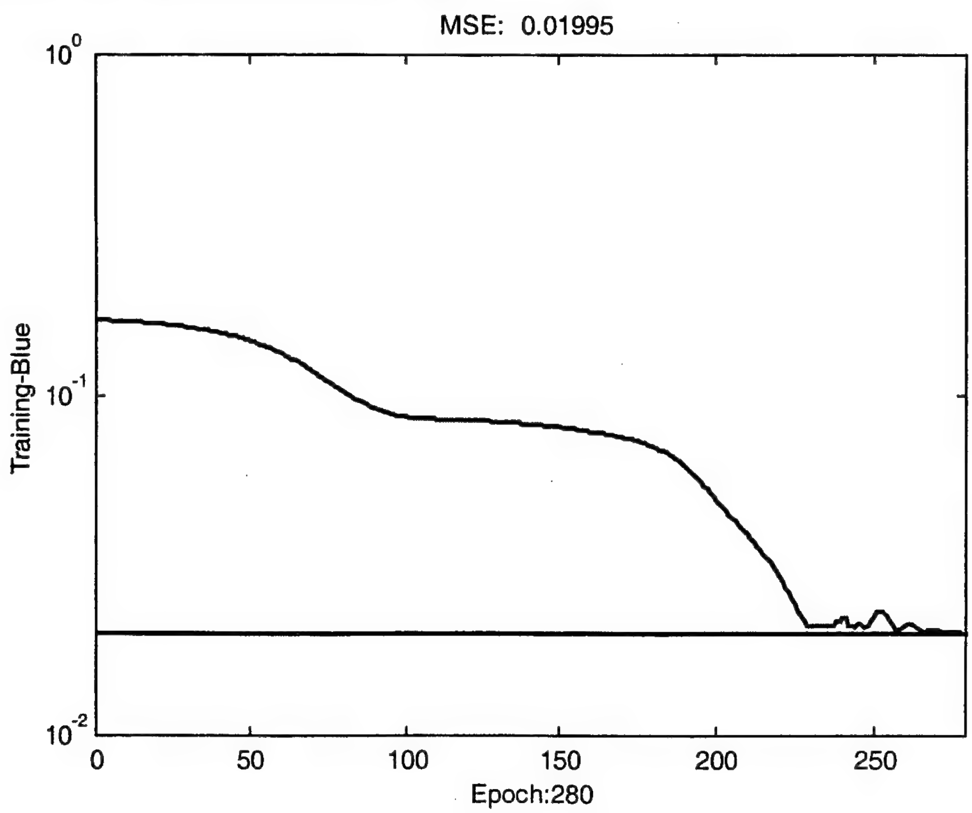
SUBJECT: NL, CBNN, FORWARD DIRECTION



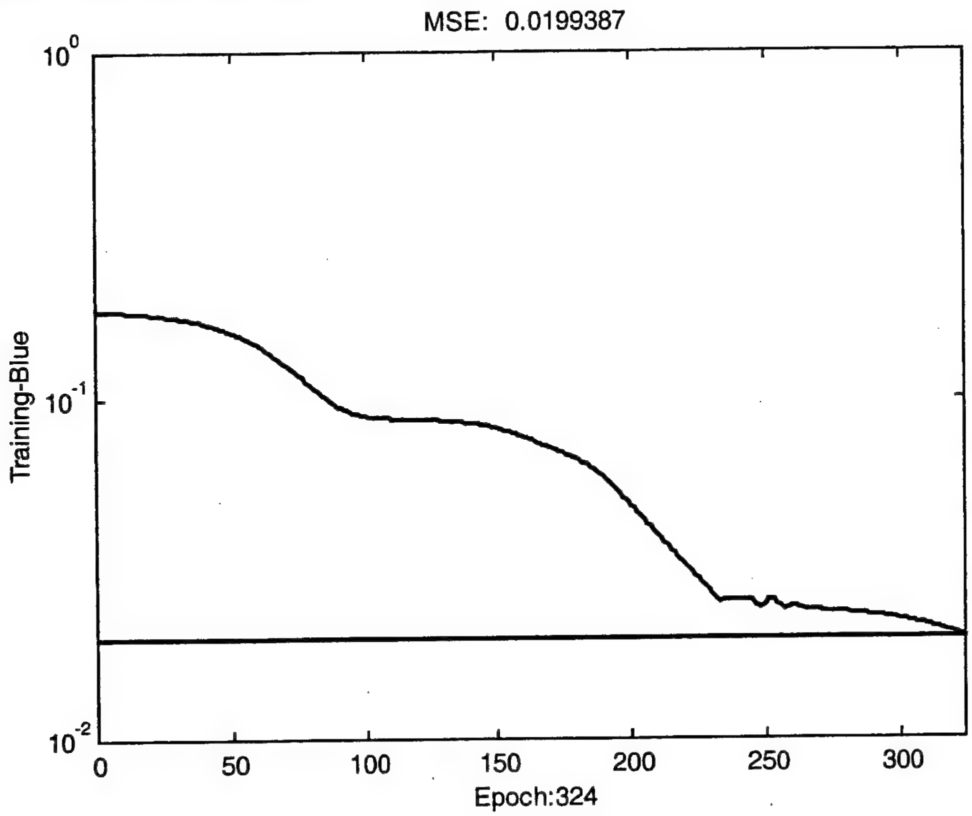
SUBJECT:AB, CBNN, BACKWARD DIRECTION



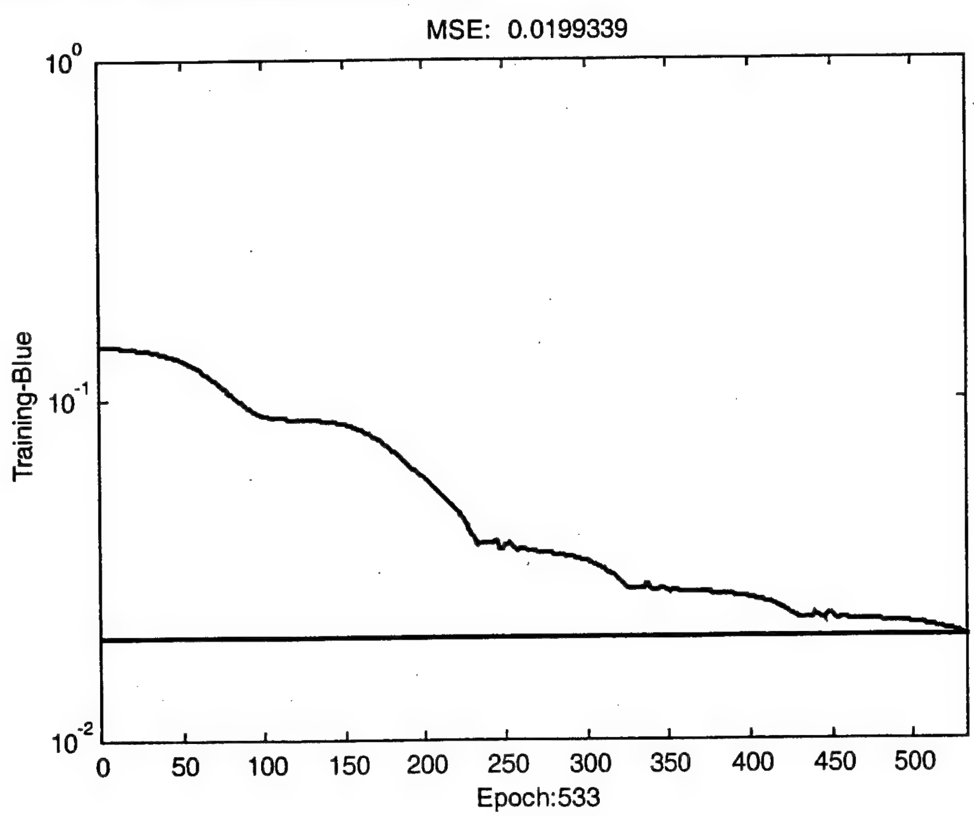
SUBJECT:BK, CBNN, REVERSE DIRECTION



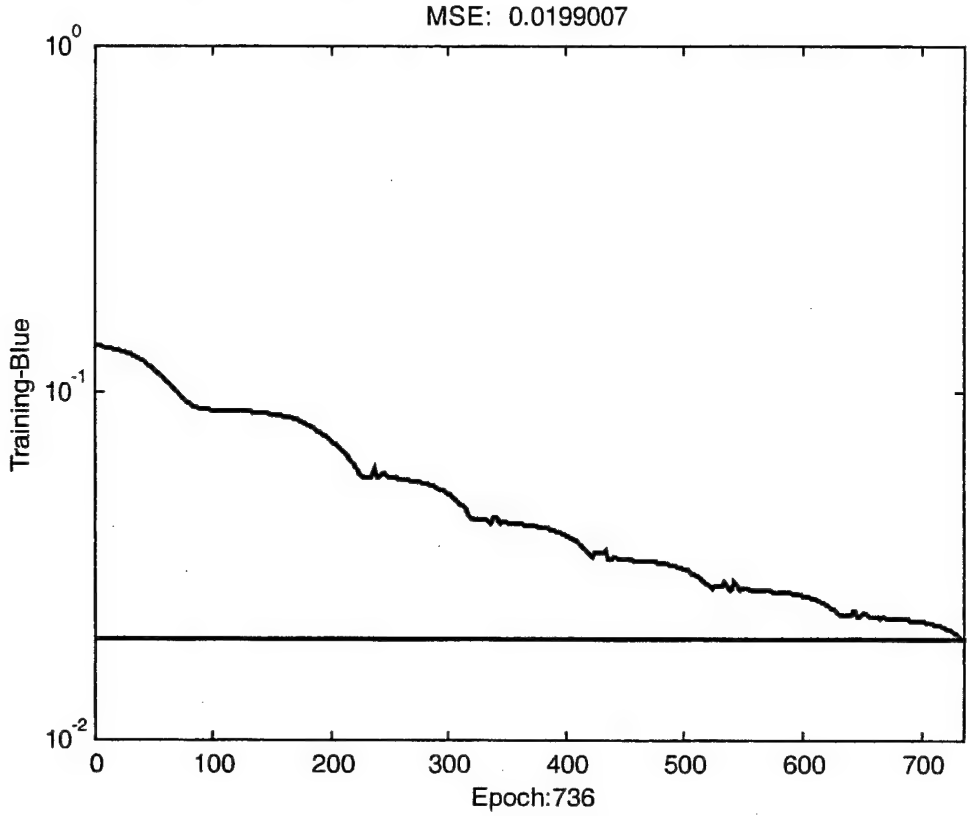
SUBJECT: ES, CBNN, REVERSE DIRECTION



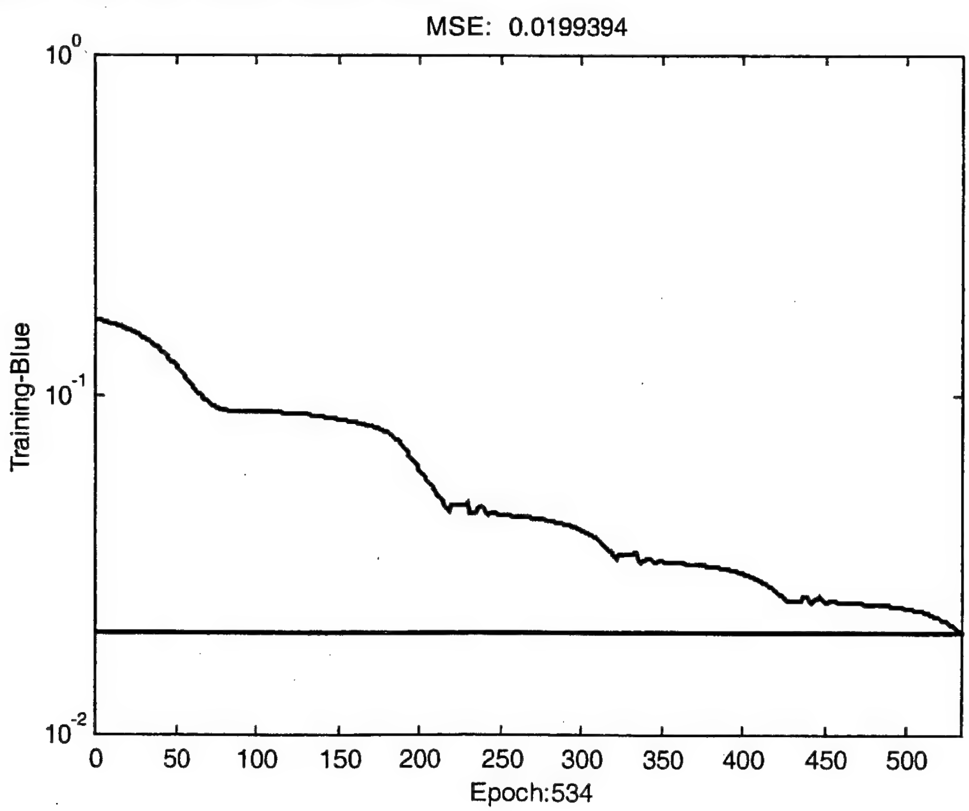
SUBJECT: NL, CBNN, REVERSE DIRECTION



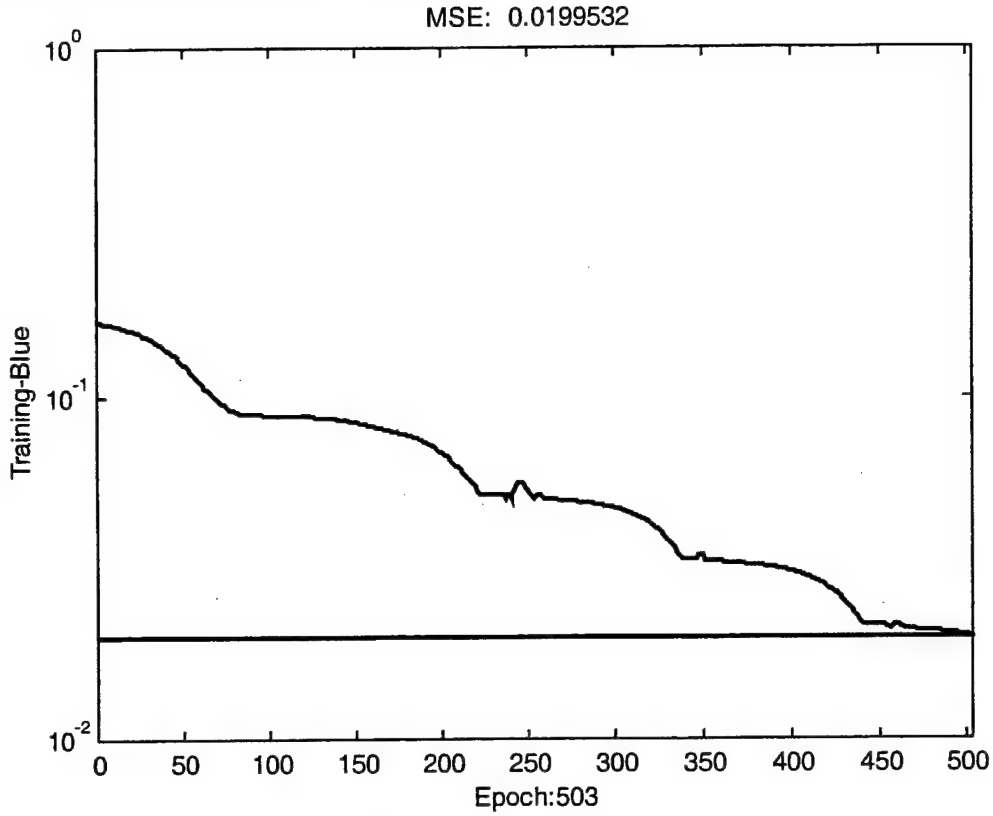
SUBJECT: AB, CBNN, DUAL DIRECTIONS



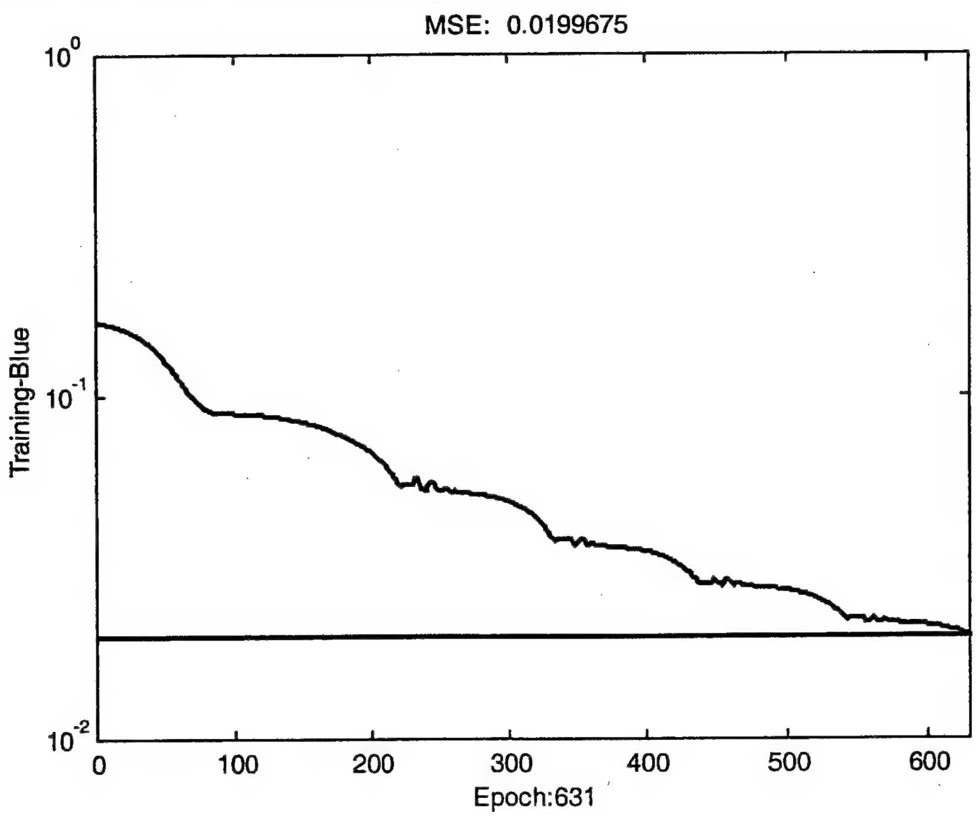
SUBJECT:BK, CBNN, DUAL DIRECTION



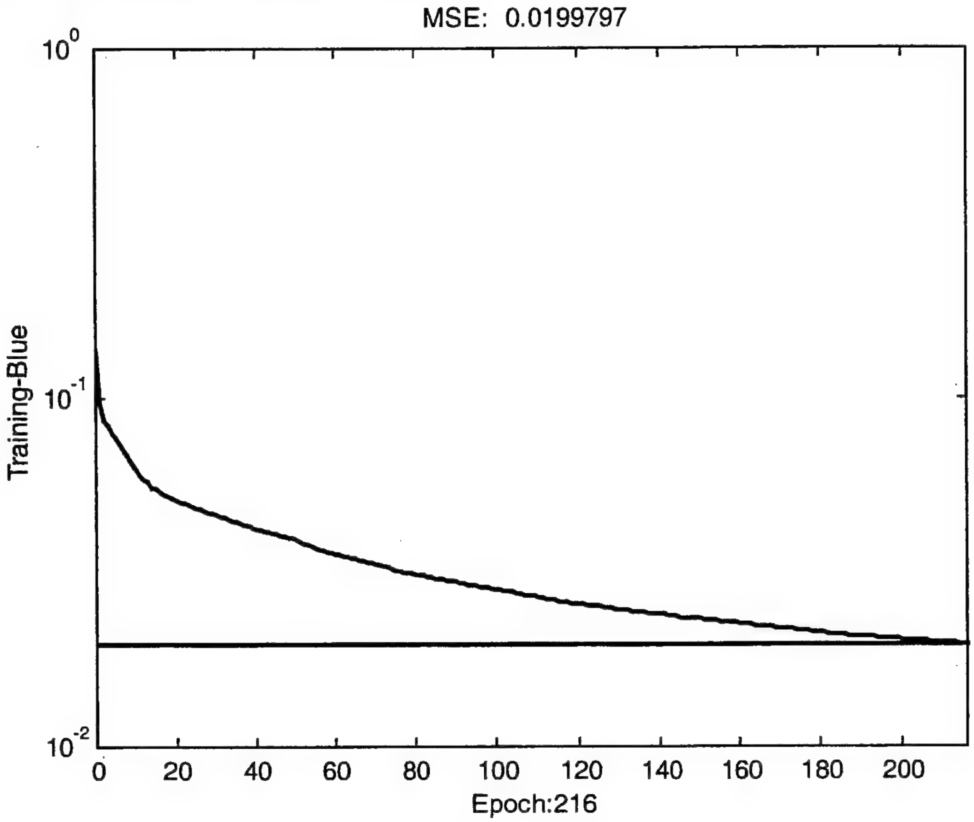
SUBJECT: ES, CBNN, DUAL DIRECTIONS



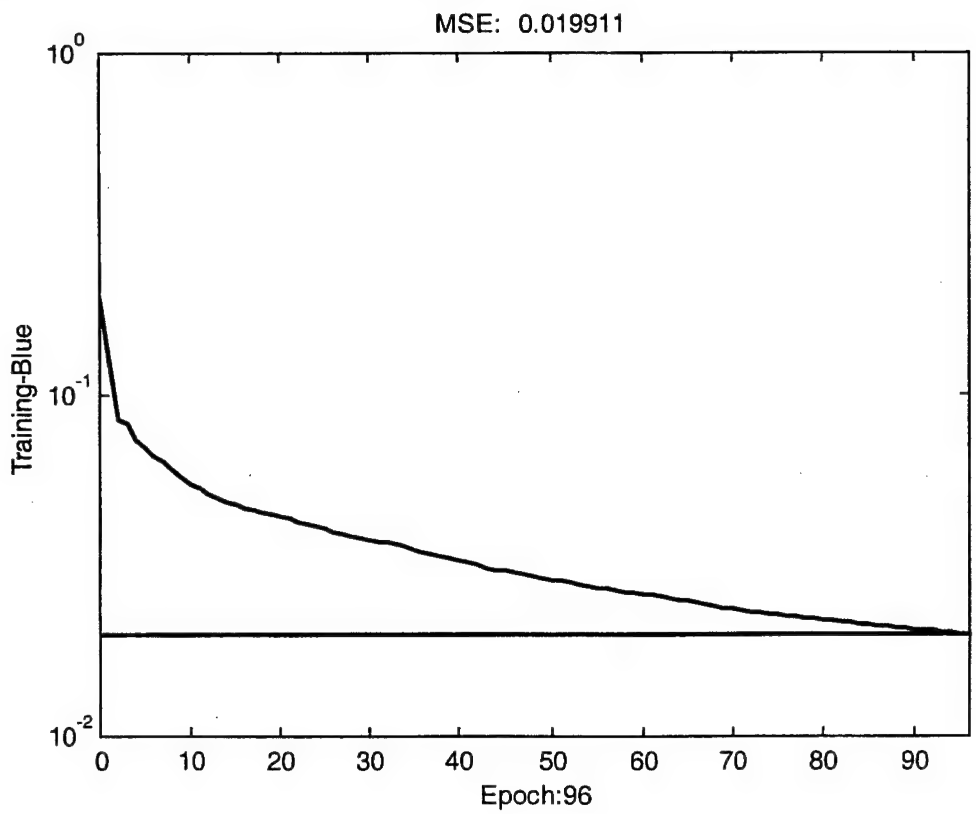
SUBJECT: NL, CBNN, DUAL DIRECTIONS



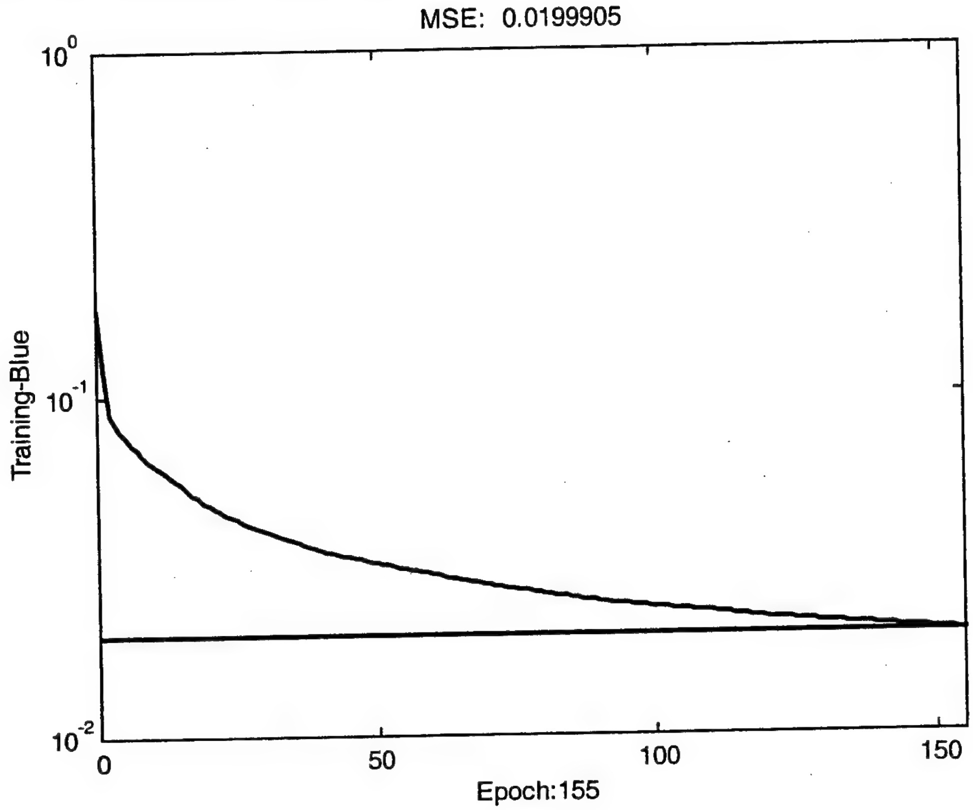
SUBJECT:AB, RBNN, FORWARD DIRECTION



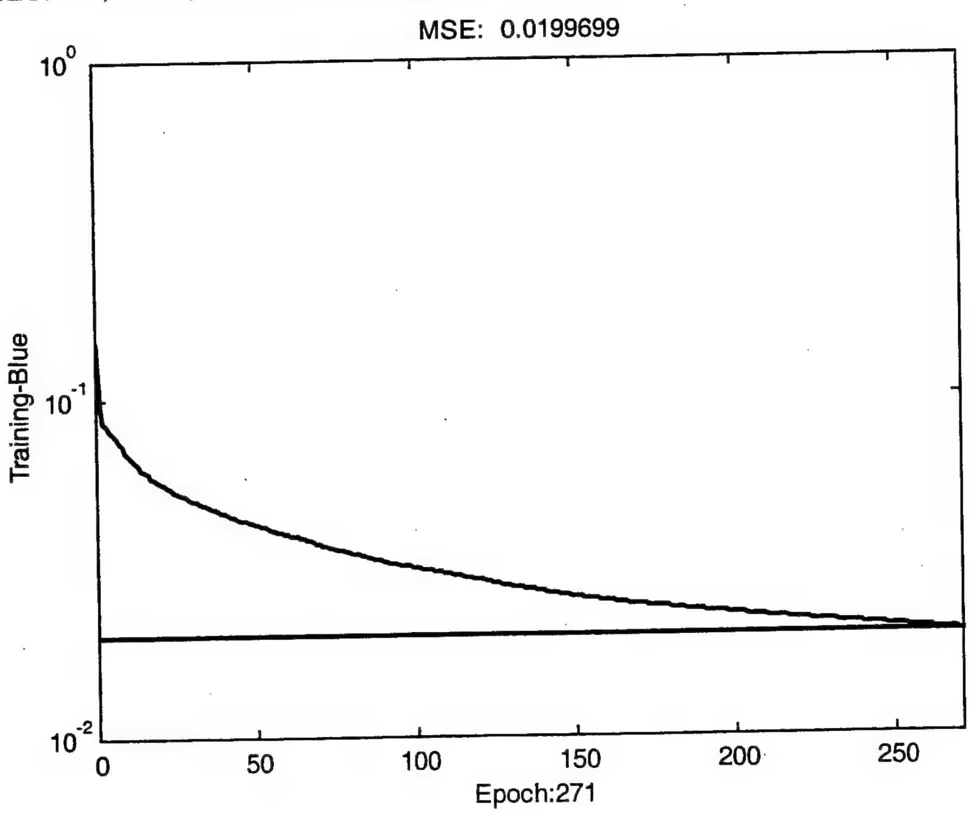
SUBJECT:BK, RBNN, FORWARD DIRECTION



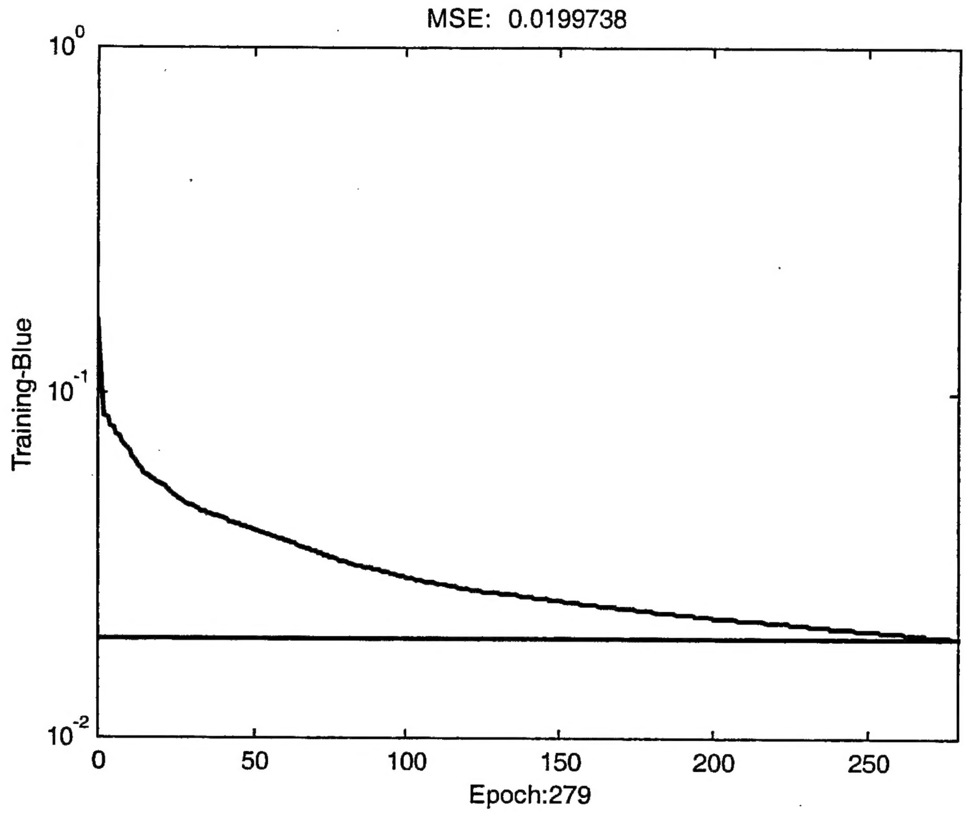
SUBJECT:ES, RBNN, FORWARD DIRECTION



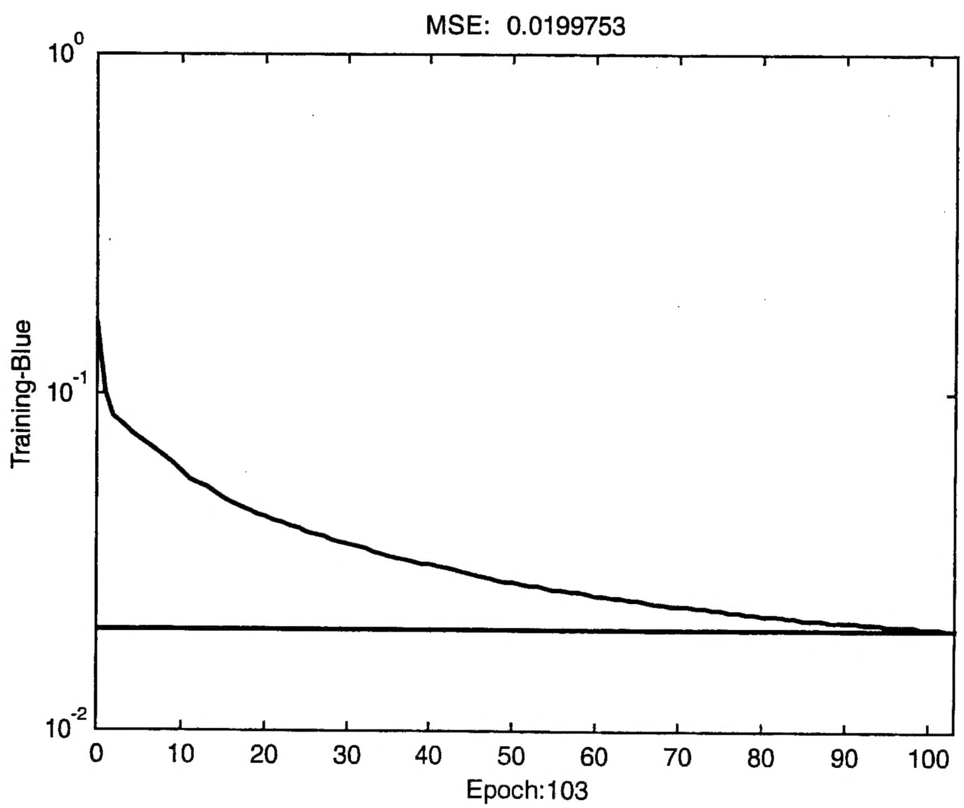
SUBJECT:NL, RBNN, FORWARD DIRECTION



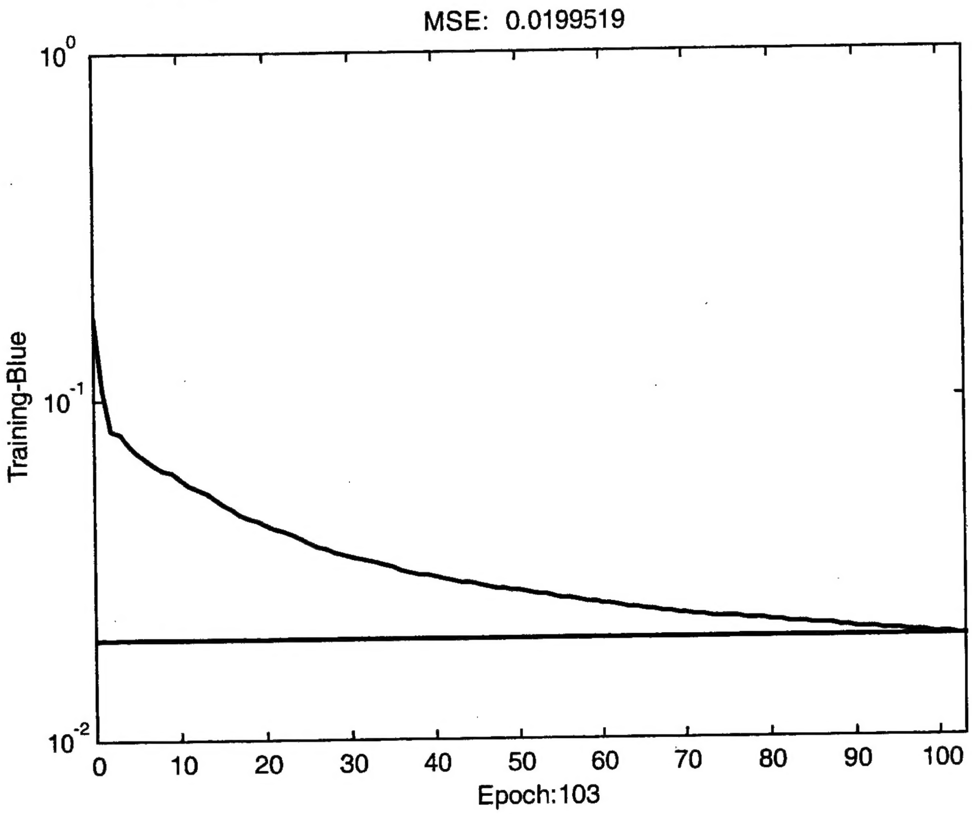
SUBJECT:AB, RBNN, REVERSE DIRECTION



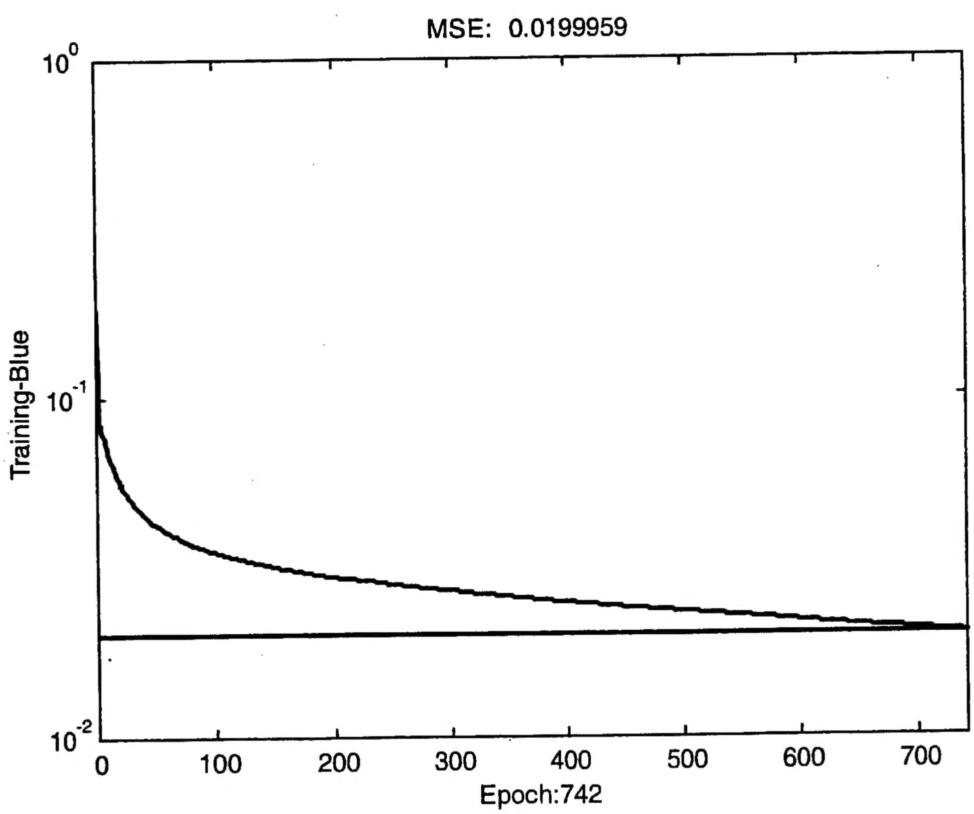
SUBJECT:BK, RBNN, REVERSE DIRECTION



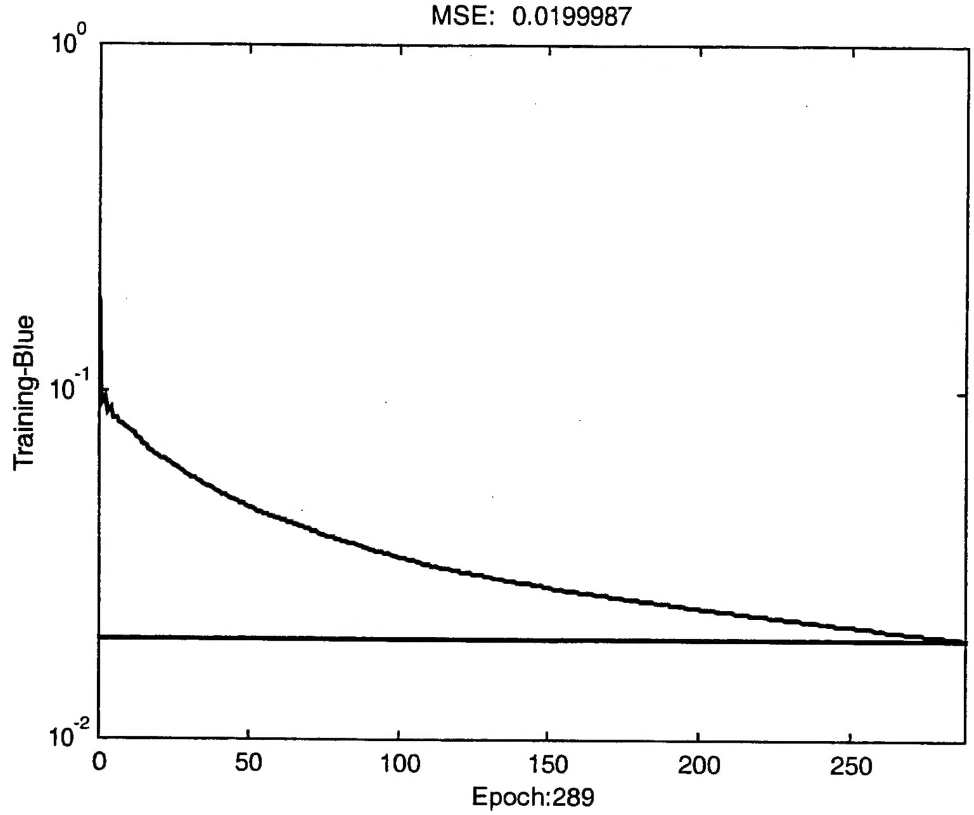
SUBJECT:ES, RBNN, REVERSE DIRECTION



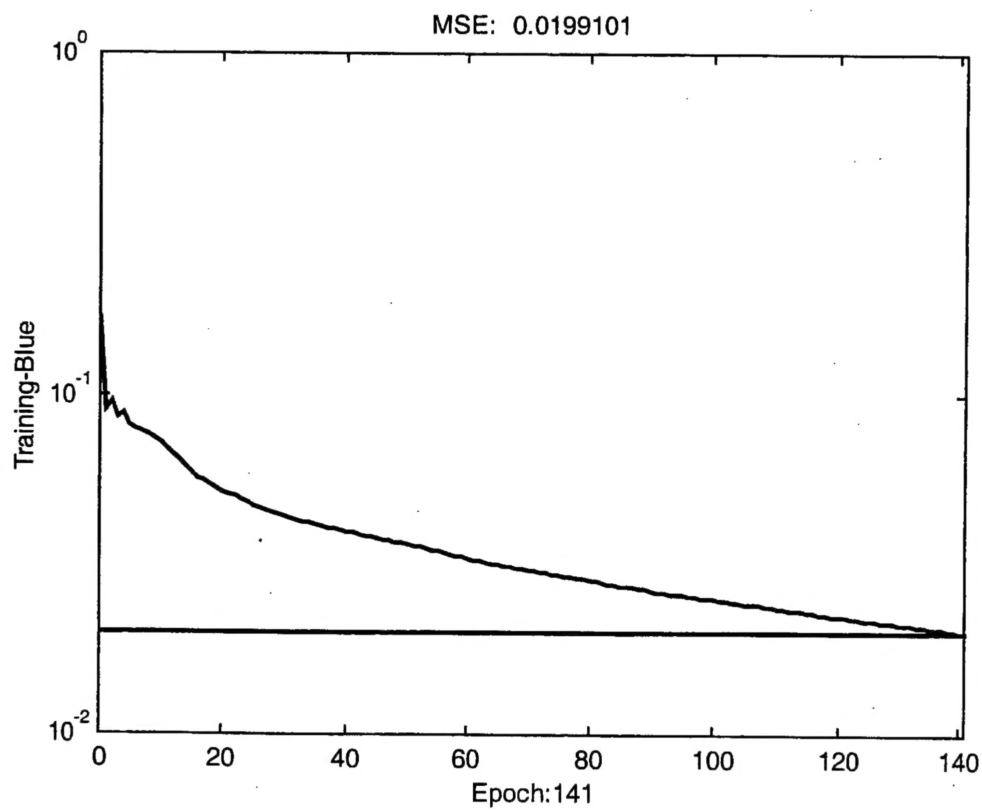
SUBJECT:NL, RBNN, REVERSE DIRECTION



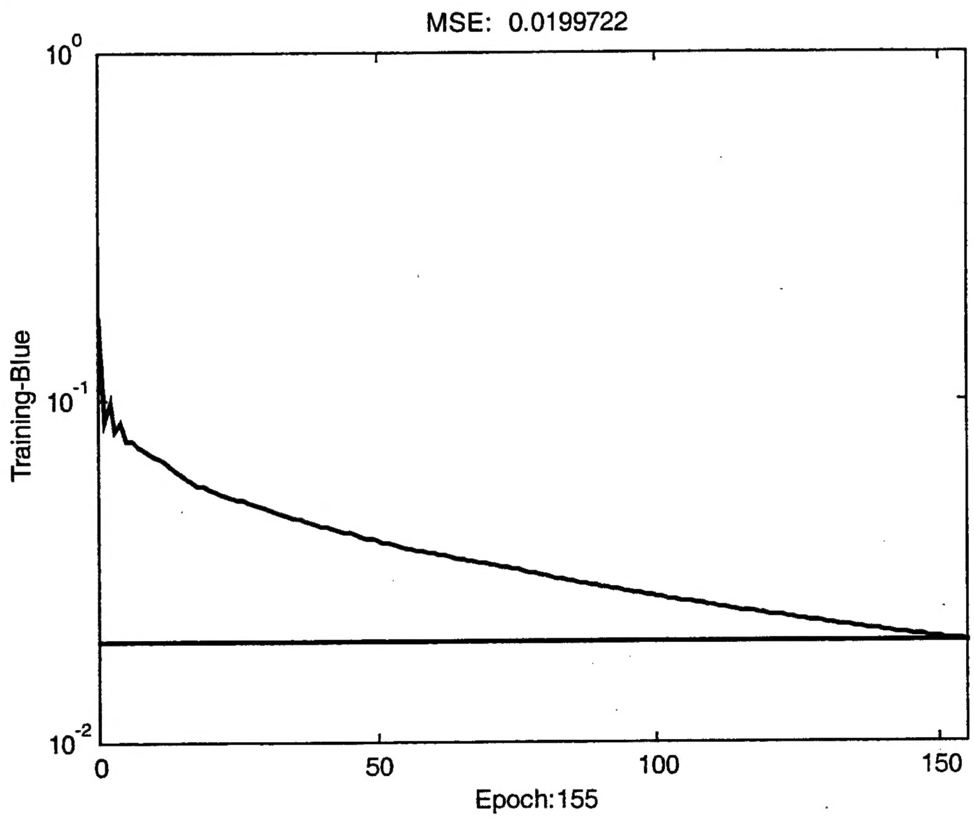
SUBJECT:AB, RBNN, DUAL DIRECTIONS



SUBJECT: BK, RBNN, DUAL DIRECTIONS



SUBJECT: ES, RBNN, DUAL DIRECTIONS



SUBJECT: NL, RBNN, DUAL DIRECTIONS